

# Beiträge zur Programmiertechnik für elektronische Analogie-Rechenmaschinen

Herrmann, Horst

Veröffentlicht in:  
Abhandlungen der Braunschweigischen  
Wissenschaftlichen Gesellschaft Band 10, 1958,  
S.117-149



Friedr. Vieweg & Sohn, Braunschweig

# Beiträge zur Programmiertechnik für elektronische Analogie-Rechenmaschinen

Von Horst Herrmann, Braunschweig<sup>1)</sup>

Vorgelegt von Herrn H. Schaefer

*Summary: The first section describes the operational units of multipurpose electronic analogue computers, diagram-symbols and transfer functions. The following sections deal with programming techniques, especially estimations of scaling factors and coefficient-settings for overload-free preliminary setups and a method to find out optimal scaling factors and time constants for integrator-chains. Solutions of highest accuracy are available with transforms of the problem into equations adapted to computer-representation. The van der POL equation is discussed as simple example for a problem with a manifold of arrangements giving solutions of different accuracy. A short demonstration of the formula for transfer functions of linear operative function units and their accuracy is given in an appendix.*

Übersicht: Abschnitt 1 enthält eine knappe Übersicht über die operativen Recheneinheiten elektronischer Analogie-Rechenmaschinen, ihre symbolische Darstellung und ihre Übertragungseigenschaften. An einem einfachen Beispiel wird in Abschnitt 2 eine Programmierskizze hergeleitet. Sodann werden Verfahren zum Ermitteln überlastungsfreier Ausgangsprogrammierungen angegeben, aus denen sich in Abschnitt 3 ein Verfahren für optimale Dimensionierung von Integriererketten ergibt. Am Beispiel der Differentialgleichung von van der POL zeigt sich, daß es bereits bei sehr einfachen Aufgaben mehrere Programmiermöglichkeiten gibt, die mit der Rechenmaschine auf Lösungen verschiedener Genauigkeit führen. Es ist daher notwendig, die gegebene Aufgabe durch Transformationen maschinengerecht umzuformen und eine umfassende Übersicht über Programmierungen für die gleiche Aufgabe zu gewinnen. Hierzu dienen Strukturskizzen mit vereinfachter Symbolik. Optimale Lösung einer Aufgabe erfordert günstigste Auswahl der Strukturskizze und optimale Programmierung für die zugehörige Programmierskizze.

Der Anhang enthält eine kurze Herleitung der Formel für die realen Übertragungseigenschaften linearer operativer Recheneinheiten.

## Vorbemerkungen

Die elektronischen Analogie-Rechenmaschinen enthalten operative Recheneinheiten und Koeffizienteneinheiten. Die Recheneinheiten elektronischer Analogie-Rechenmaschinen können auf mannigfache Weise zu Netzwerken zusammengeschaltet werden. An den Ausgängen der operativen Recheneinheiten ergeben sich dann zeitlich veränderliche Spannungen, die registriert werden können. Dies geschieht auf Bildschirmen von Kathodenstrahlröhren oder mit schreibenden Registriergeräten.

Programmieren für eine gegebene Aufgabe erfordert Entwerfen und Zusammenschalten eines Rechnernetzwerkes, in dem die gewünschten Funktionen (in geeignetem Funktionsmaßstab als Spannungen in Abhängigkeit von der Zeit) abgegriffen werden können. Die Aufgabe kann darin bestehen,

<sup>1)</sup> Erweiterte Fassung des am 13. 9. 57 während der Tagung der Deutschen Mathematiker-Vereinigung in Dresden gehaltenen Vortrags.

eine Funktion oder etliche Funktionen einer Veränderlichen zu ermitteln. Diese Funktionen können formal als Lösungen von Differentialgleichungen oder inhaltlich durch Blockschemata von Übertragungseigenschaften bestimmt sein.

Die Genauigkeit, mit der die Lösungen der gegebenen Aufgabe durch die Ausgangsspannungen von Recheneinheiten dargestellt werden können, ist begrenzt. Sie ist wesentlich von der Programmierung abhängig. Daher entstehen beim Rechnen mit Analogie-Maschinen die Aufgaben,

1. möglichst große Genauigkeit zu erreichen,
2. die erhaltene Genauigkeit zu beurteilen.

Im folgenden wird vorwiegend die erste dieser Aufgaben erörtert. Zum leichteren Verständnis wird an einige Eigenschaften der Recheneinheiten erinnert und die verwendete Symbolik erläutert.

### 1. Die Recheneinheiten elektronischer Analogie-Rechenmaschinen

Die linearen operativen Recheneinheiten der meisten elektronischen Analogie-Rechenmaschinen enthalten einen Gleichspannungsverstärker mit hoher Verstärkung (Abb. 1a). Für ideale Verstärker (mit unendlicher Verstärkung) gilt nach [1], [2], [3], [4]<sup>1)</sup> und Anhang S. 148: Wird eine von der Zeit  $\tau$  abhängige Spannung  $v(\tau)$  über einen Widerstand  $R$  an den Eingang

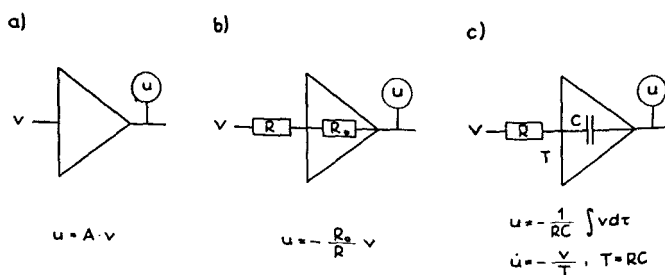


Abb. 1. a) Verstärker, b) Umkehrer, c) Integrierer

des Verstärkers gelegt und die Ausgangsspannung  $u(\tau)$  durch einen Widerstand  $R_0$  an den Eingang rückgekoppelt (Abb. 1b), so stellt sich zwischen Eingangs- und Ausgangsspannung die Beziehung

$$u = - \frac{R_0}{R} v \quad (1.1)$$

ein. Erfolgt die Rückkopplung mit einem Kondensator der Kapazität  $C$  (Abb. 1c), so erhält man zwischen der Eingangsspannung  $v(\tau)$  und der Ausgangsspannung  $u(\tau)$  die Beziehung

$$\dot{u} = - \frac{v}{RC}, \quad u = - \frac{1}{RC} \int_0^{\tau} v d\tau + u_0. \quad (1.2)$$

<sup>1)</sup> Hinweis auf das Literaturverzeichnis S. 35.

In (1.2) und im folgenden ist die Ableitung nach der Zeit  $\tau$  durch einen Punkt bezeichnet, und  $u_0 = u(0)$  ist der Wert der Ausgangsspannung beim Einschalten der Recheneinheit zur Zeit  $\tau = 0$ .

Für reale Verstärker mit endlicher frequenzabhängiger Verstärkung gelten (1.1) und (1.2) nur, sofern die Ausgangsspannung  $u(\tau)$  im Bereich der zulässigen Belastbarkeit liegt:  $-U \leq u \leq U$  (beispielsweise  $U = 50$  Volt oder  $U = 100$  Volt). In diesem Bereich sind (1.1) und (1.2) mit einer von der Verstärkung, vom Frequenzspektrum der Funktion  $v(\tau)$  und von der Rechenzeit abhängigen Genauigkeit als **Programmiersätze** anwendbar, (1.1) für Multiplizieren mit einem konstanten Faktor, (1.2) für Integrieren nach der Zeit. Die hierdurch gekennzeichneten Genauigkeitsfragen sind in der Literatur mehrfach diskutiert worden [1], [4], [5], [6]. Eine kurze Herleitung der realen Übertragungseigenschaften linearer operativer Recheneinheiten ist im Anhang dargestellt (S. 148).

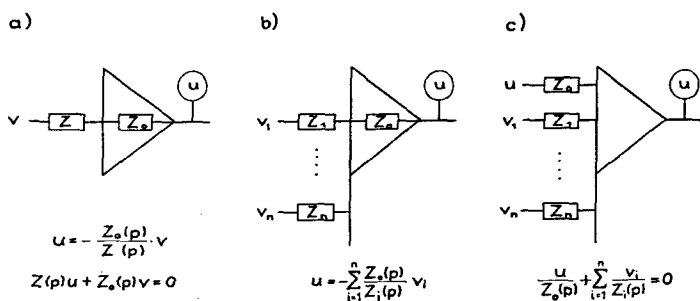




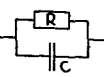
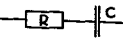
Abb. 2. Beispiele für lineare operative Recheneinheiten und ihre Übertragungseigenschaften

Allgemeiner als nach Abb. 1 werden lineare operative Recheneinheiten erhalten, indem man die Eingangsspannung  $v(\tau)$  über eine passive Impedanz  $Z$  ankoppelt und die Ausgangsspannung mit einer passiven Impedanz  $Z_0$  rückkoppelt (Abb. 2a). Schließlich können dem gleichen Verstärker mehrere Eingangsspannungen  $v_1, v_2, \dots, v_n$  durch Impedanzen  $Z_1, Z_2, \dots, Z_n$  zugeführt werden. Solche Schaltungen werden nach Abb. 2b, c symbolisch dargestellt. Mit idealen Verstärkern erhält man hierfür die in Abb. 2 angeschriebenen Programmiersätze (Herleitung im Anhang). Als Impedanzen kommen Zweipole und Dreipole in Betracht. Hierbei beschränkt man sich meistens auf Netzwerke aus Widerständen und Kondensatoren, weil diese quantitativ leichter mit größter Genauigkeit realisiert werden können als Selbstinduktionen. Die am häufigsten verwendeten Zweipole und ihre Impedanzen sind in Abb. 3a zusammengestellt. Besonders reichhaltige Tabellen (Zweipole, Dreipole und Übertragungseigenschaften von Recheneinheiten) findet man in [1] und in [7].

Zur analytischen Darstellung der Impedanzen und der Übertragungseigenschaften von Recheneinheiten wird das Zeichen  $p$  mit der folgenden Vereinbarung verwendet.  $Z(p)$  ist ein rationaler Ausdruck in  $R, C, p$ . Aus  $u = -\frac{Z_0(p)}{Z(p)} v$  entsteht durch Einsetzen

von  $Z_0(p)$  und  $Z(p)$  zunächst ein Ausdruck  $u = -\frac{F(p)}{G(p)}v$ , in dem  $F(p)$  und  $G(p)$  Polynome in  $p$  sind.  $G(p)u(\tau) + F(p)v(\tau) = 0$  ist dann in  $p$  ganz rational. Durch Einsetzen von  $p = \frac{d}{d\tau}$  entsteht hieraus eine lineare Differentialgleichung mit konstanten Koeffizienten.  $u(\tau)$  ist eine Lösung dieser Differentialgleichung mit der gegebenen (Eingangs-) Funktion  $v(\tau)$ . In den linearen operativen Recheneinheiten allgemeiner elektronischer Analogie-Rechenmaschinen werden außer reinen Widerstandsnetzwerken nur Impedanzen verwendet, mit denen diese Differentialgleichung linear von erster Ordnung wird. Ihre Lösung  $u(\tau)$  ist daher durch die Koeffizienten — also durch Werte für  $R$  und  $C$  — und durch den Anfangswert  $u(0)$  für  $u(\tau)$  am Ausgang der Recheneinheit beim Einschalten zur Zeit  $\tau = 0$  eindeutig bestimmt. Für Abb. 2c ergibt sich hiernach zunächst  $u = -\frac{1}{RCp}v$ ,

und dies bedeutet  $RCpu + v = 0$  oder  $RC\dot{u} + v = 0$  oder  $\dot{u} = -\frac{v}{RC} = -\frac{v}{T}$ , wenn  $RC = T$  gesetzt wird. Durch Integrieren folgt (1.2). In speziellen elektronischen Analogie-Rechenanlagen werden auch allgemeinere passive Netzwerke verwendet. Dann können mit einem Verstärker Übertragungsbeziehungen erhalten werden, die durch Differentialgleichungen höherer als erster Ordnung zu beschreiben sind. Im folgenden werden nur allgemeine elektronische Analogie-Rechenmaschinen berücksichtigt.

a)	Nr.	Schaltsymbol	Z
	1		$R$
	2		$\frac{1}{Cp}$
	3		$\frac{R}{1+Tp}$
	4		$\frac{1+Tp}{Cp}$

Abkürzung:  $T = RC$

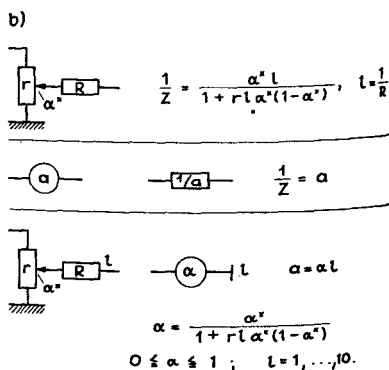


Abb. 3. a) Rechenimpedanzen, b) Konstante Koeffizienten

**Konstante Koeffizienten** können, wie aus Abb. 2 und Abb. 3b hervorgeht, durch Widerstände oder Netzwerke von Widerständen dargestellt werden, da deren Impedanzen konstant sind. Beispielsweise liefert ein lineares Potentiometer (Spannungsteiler) mit Widerstand  $r$  und Anschluß-Widerstand  $R$  (Leitfähigkeit  $l = \frac{1}{R}$ ) nach Abb. 3b eine Impedanz, deren reziproker Wert (bis auf eine Belastungskorrektur) zum Einstellwert  $\alpha^*$  des Potentiometers proportional ist. Koeffizienteneinheiten werden symbolisch durch einen Kreis mit eingeschriebenem Einstellwert (Zahl, Buchstabe oder Formel) dargestellt. In den elektronischen Analogie-Rechenmaschinen sind zwei wesentlich zu unterscheidende Realisierungen der Koeffizienten-Einstellwerte gebräuchlich:

a) **Potentiometer-Koeffizienteneinheiten**: Der Einstellwert  $a$  wird zerlegt in ein Produkt  $a = \alpha \cdot l$  aus einem am Potentiometer kontinuierlich einstellbarem

Wert  $0 \leq \alpha \leq 1$  und einem in diskreten Stufen verfügbaren Anschlußleitwert  $l$ , beispielweise  $l = 1, 5, 10$  oder  $l = 1, 4, 10$  oder  $l = 1, 2, 3, \dots, 10$ . Bis zu 40 Potentiometer werden auf einer gemeinsamen Frontplatte zu einer Potentiometer-Matrix zusammengefaßt. Die Einstellungen von Potentiometer-Koeffizienteneinheiten erfordern Belastungskorrekturen (vgl. Abb. 3b). Diese können nach Nomogrammen ermittelt oder genauer durch Brücken-Vergleichsmessungen kompensiert oder durch einen Servomechanismus automatisch ausgeführt werden<sup>1)</sup>.

b) *Widerstände oder Widerstandsnetzwerke* (hochohmig): Der Einstellwert  $a$  wird durch die Impedanz  $R = \frac{1}{a}$  realisiert. Dabei wird  $R$  dekadisch mit drei zählenden Ziffern und verschieblichem Komma eingestellt. Abb. 4 zeigt zwei

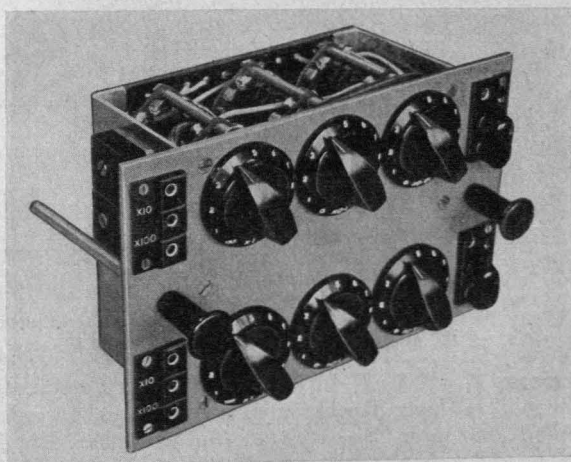


Abb. 4. Koeffizientenpaar mit dreiziffriger Einstellung,  $N = n_1, n_2, n_3$ ,  $1,00 \leq N \leq 10,99$  in Stufen 0,01 und mit Kommaeinstellung durch Kurzschlußstecker,  $R = kN$ ,  $k = 0,1 \ 1 \ 10 \ 100$ . Foto Short Brothers and Harland

hierauf beruhende Koeffizienteneinheiten, die auf der gleichen Frontplatte übereinander angeordnet sind. Belastungskorrektur ist nicht notwendig. Für Einstellungen, die größere Genauigkeit als 0,5% erfordern, kann die Einstellungsgenauigkeit korrigiert werden.

In beiden Fällen a) und b) sind die Einstellwerte  $0,001 \leq a \leq 10$  verfügbar. Der Bereich  $0,1 \leq a \leq 10$  werde als Hauptbereich bezeichnet.

Die *linearen operativen Recheneinheiten* sind nach dem Baukastenprinzip in Einschübe montiert, die einen oder zwei Verstärker enthalten mitsamt einer

<sup>1)</sup> Beispiel: Bei Einstellung des Abgriffes in der Mitte des Potentiometers ist  $\alpha^* = 0,5$ . Ist der Widerstand des Potentiometers  $r = 50 \text{ k}\Omega = 0,05 \text{ M}\Omega$  und der Anschlußleitwert  $l = 10$  ( $R = 0,1 \text{ M}\Omega$ ), so ergibt die in Abb. 3b angemarkte Formel für den mit  $\alpha^* = 0,5$  eingestellten Koeffizientenfaktor  $\alpha$  den Wert  $\alpha = \frac{0,5}{1 + 0,05 \cdot 10 \cdot 0,5 \cdot 0,5} = \frac{0,5}{1,125} = 0,444 \dots$  Die Abweichung zwischen  $\alpha^*$  und  $\alpha$  ist die Belastungskorrektur.

mehr oder weniger umfangreichen Ausstattung mit Übertragungseigenschaften. Zwei typische Beispiele mit je einem Verstärker werden näher beschrieben.

a) Abb. 5 zeigt den Telefunken-Rechenverstärker I, Vrk 463/1, mit erläuternder Schaltskizze. Links sind Buchsen für Anschluß an Leitwerte

$l = \frac{1}{R} = 1, 1, 1, 4, 4, 10, 10, \infty$  angeordnet, rechts 5 Buchsen für Anschluß

des Ausganges und je eine Buchse für Bezugspotential (0), direkten Eingangsanschluß und Anschließen von Anfangswerten für Integrieren. Mit dem Schalter wird die Übertragungseigenschaft gewählt: In Stellung  $\Sigma$  wird die Einheit zum Summierer durch Rückkopplung mit  $R = 1 M\Omega$ , in Stellung  $\int$

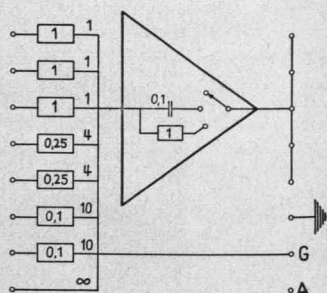
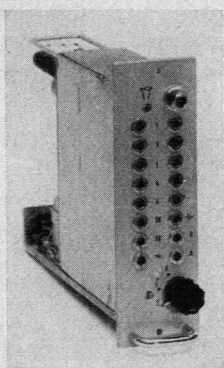


Abb. 5. Lineare operative Recheneinheit TELEFUNKEN Vrk 463/1, links Einschubeinheit, rechts Schaltsymbol. Pressefoto Telefunken

zum Integrierer mit  $C = 0,1 \mu F$ . In Stellung  $V$  ist der Verstärker frei verfügbar, so daß er mit beliebigen Rückkopplungen überbrückt werden kann; insbesondere sind Kapazitäten  $C = 1 \mu F$  hierfür verfügbar. Die Eingangsleitwerte werden entweder unmittelbar oder für Anschließen von Potentiometer-Koeffizienteneinheiten verwendet; diese Widerstände können auch zueinander und zur Rückkopplung parallel geschaltet werden.

b) Abb. 6 zeigt eine lineare operative Recheneinheit, die reichlicher mit Übertragungseigenschaften ausgestattet ist. Mit zwei Schaltern sind Widerstandswerte wählbar,

$$R_1, R_2 = 0,01 \ 0,02 \ 0,05 \ 0,1 \ 0,2 \ 0,5 \ 1 \ \infty M\Omega,$$

mit einem weiteren Schalter sind Kapazitäten wählbar,

$$C = 0 \ 0,001 \ 0,003 \ 0,01 \ 0,03 \ 0,1 \ 0,3 \ 1 \mu F,$$

die durch einen Übertragungswahlschalter zu sieben Paaren von Eingangs- und Rückkopplungsimpedanzen zusammengeschaltet werden können. Es sind dies die Paarungen (1,1), (1,2), (1,3), (1,4), (4,1), (3,1), (2,1) der in Abb. 3a dargestellten Impedanzen. In einer solchen Recheneinheit sind daher in diskreter Auswahl, teils auf mehrfache Weise einstellbar, 19 konstante Faktoren  $k = \frac{R_2}{R_1}$  im Bereich  $10^{-2} \leq k \leq 10^2$  und 29 Zeitkonstanten  $T = RC$  im Bereich  $10^{-5} \leq T \leq 1$  verfügbar, so daß sich insgesamt 889 verschieden dimen-

sionierte Übertragungen ergeben. Durch die Schalterstellungen an den Grenzen der Bereiche ist der Verstärker frei verfügbar, besonders zum Zusammenschalten mit Koeffizientendekaden nach Abb. 4. Jede solche Einheit ist überdies mit einer Anfangswert-Eingabe und einer Überlastungsanzeige ausgestattet. Diese wird durch ein Thyatron eingeschaltet und zeigt daher beliebig kurzzeitige Überlastung der Recheneinheit verlässlich an.

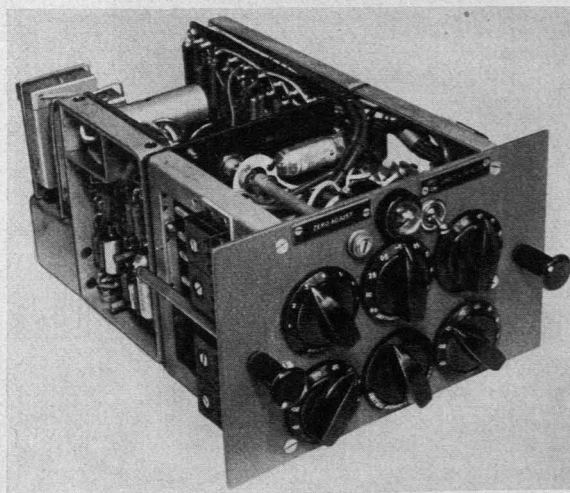


Abb. 6. Lineare operative Recheneinheit des SHORTS GENERAL PURPOSE ANALOGUE COMPUTER mit Schaltern für Wahl und Dimensionierung der Übertragung, Grob- und Feinstellung für Anfangswerte. Foto Short Brothers and Harland

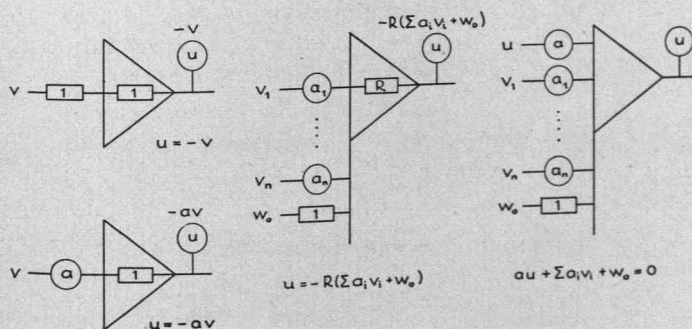


Abb. 7. Vorzeichenumkehrung, konstanter Faktor, Summieren von Funktionen

Die in den Programmierskizzen für Rechnernetzwerke am häufigsten anzuwendenden Darstellungen für Schaltungen von linearen operativen Recheneinheiten in Verbindung mit Koeffizienteneinheiten sind in Abb. 7 für Summierer und in Abb. 8 für Integrierer zusammengestellt. Bei den Integrierern gibt es für das Einführen von Anfangswerten zwei Möglichkeiten (Abb. 9): Man kann den Kondensator des Integrierers vor Beginn der Rechnung durch



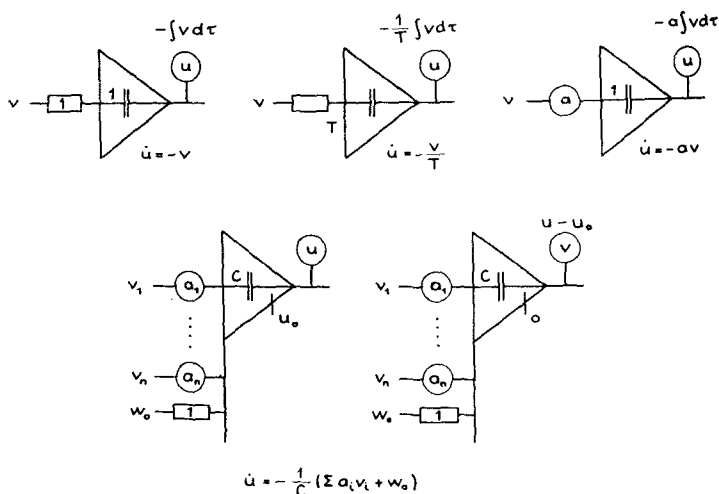


Abb. 8. Integrieren einer Funktion, Integrieren von Summen

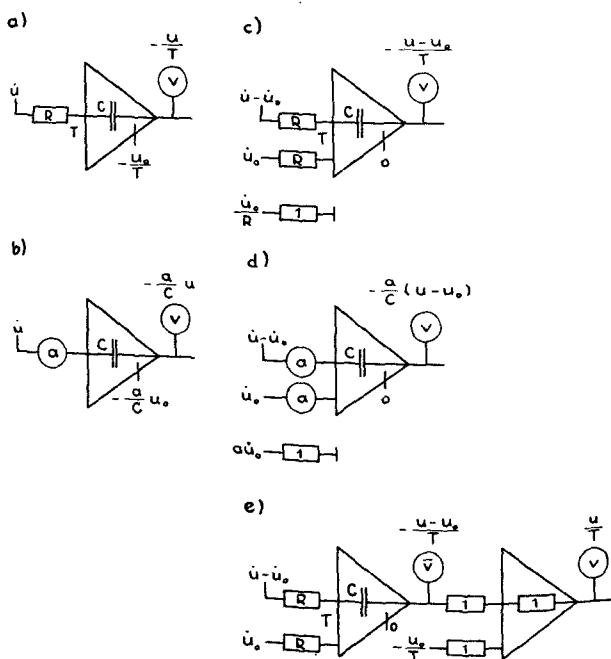
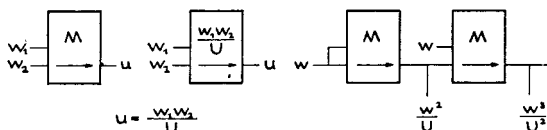


Abb. 9. Eingabe der Anfangswerte, a), b) am Integrator selbst, c), d), e) in die nachfolgenden Recheneinheiten

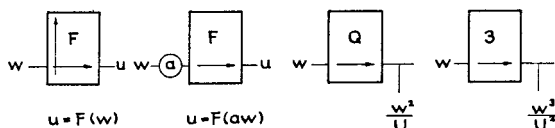
eine Spannungsquelle aufladen, die zu Beginn der Rechnung abzuschalten ist (Abb. 9a, b). Statt dessen kann man die Integration mit dem Anfangswert 0 beginnen lassen und die geforderten Anfangswerte in die nachfolgenden Recheneinheiten einführen (Abb. 9c, d, e). Dies geschieht durch Einschalten einer Sprungfunktion zu Beginn der Rechnung. Nur ausnahmsweise ist bei Anwendung dieses Verfahrens ein besonderer Summierer notwendig (Abb. 9e).

### Kontinuierliche nichtlineare Recheneinheiten

#### MULTIPLIZIERER

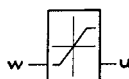


#### FUNKTIONSERZEUGER



### Diskontinuierliche Recheneinheiten

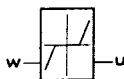
#### BEGRENZER



$$\begin{aligned} w &\leq \underline{w} & u &= \underline{u} \\ \underline{w} &\leq w \leq \bar{w} & u &= kw \\ w &> \bar{w} & u &= \bar{u} \end{aligned}$$

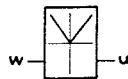
$$\begin{aligned} u &= \underline{u} \\ u &= kw \\ u &= \bar{u} \end{aligned}$$

#### TOTE ZONE



$$\begin{aligned} u &= k_1(w - \underline{w}) \\ u &= 0 \\ u &= k_2(w - \bar{w}) \end{aligned}$$

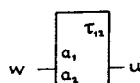
#### BETRAG



$$u = k \cdot |w|$$

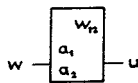
### Diskontinuierliche Schalteinheiten

#### ZEITSCHALTUNG



$$u = \begin{cases} a_1 w & 0 \leq \tau \leq \tau_{12} \\ a_2 w & \tau_{12} < \tau \leq \bar{\tau} \end{cases}$$

#### FUNKTIONSSCHALTUNG



$$u = \begin{cases} a_1 w & w < w_{12} \\ a_2 w & w \geq w_{12} \end{cases}$$

Abb. 10. Nichtlineare Recheneinheiten

Programmiersymbole für *nichtlineare Recheneinheiten* mitsamt ihren Übertragungseigenschaften sind in Abb. 10 zusammengestellt. Die wichtigsten kontinuierlichen nichtlinearen Recheneinheiten sind Multiplizierer, allgemeine Funktionsgeneratoren und spezielle Funktionsgeneratoren wie Quadrierer und Sinus/Cosinus-Generator. Es gibt vollständige und unvollständige Multiplizierer; die unvollständigen erfordern Zuschalten von Vorzeichen-Umkehrern,

sofern nicht die Faktoren mit beiden Vorzeichen im Rechnernetzwerk verfügbar sind.

Die diskontinuierlichen Recheneinheiten, „Begrenzer“, „Tote Zone“ und „Betrag“, sind spezielle Funktionserzeuger, die — zusammen mit Funktionseinheiten — auch Darstellung zusammengesetzter Diskontinuitäten (wie z. B. Hystereseschleifen) gestatten. Schließlich gibt es Schalteinheiten, mit

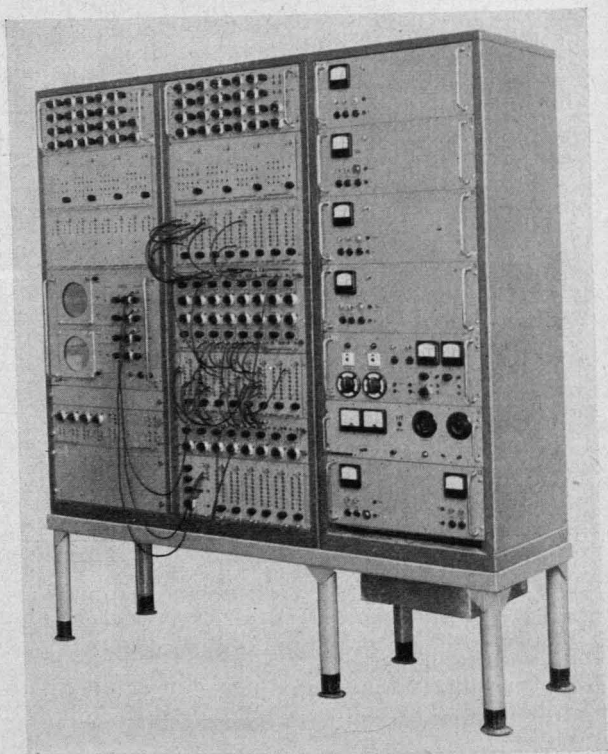


Abb. 11. Gesamtansicht des TELEFUNKEN-Analogrechners (Labormodell). Links: Funktionsgenerator, 4 Dioden-Multiplizierer, 8 Umkehrer, Sichtgerät, diskontinuierliche Recheneinheiten.

Mitte: Funktionsgenerator, 4 Dioden-Multiplizierer, 8 lineare Recheneinheiten, Potentiometerfeld für 16 Koeffizienten, weitere 8 lineare Recheneinheiten, 8 Potentiometer, Schaltfeld für repetierendes Rechnen und weitere 6 lineare Recheneinheiten. Rechts: Spannungsversorgung und Schaltfeld. Pressefoto Telefunken

denen Koeffizienten während der Rechnung sprunghaft geändert werden können, entweder nach vorgegebener Rechenzeit oder sobald eine im Rechnernetzwerk auftretende Funktion einen vorgegebenen Wert erreicht.

Die Rechen- und Koeffizienteneinheiten werden nach dem Baukastenprinzip als Einschübe, weitgehend gegeneinander austauschbar, in Gehäusen untergebracht, in denen meistens die Spannungsversorgung enthalten ist. Die Abbildungen 11, 12, 13 zeigen typische handelsübliche elektronische Analogie-Rechenmaschinen.

Die Ein- und Ausgänge aller operativen Recheneinheiten sowie auch die Anschlüsse aller Koeffizienteneinheiten sind an Buchsenfeldern verfügbar. Das Zusammenschalten zu einem Rechnernetzwerk erfolgt in der Anlage Abb. 11 durch Kabelverbindungen über die Frontplatte hinweg, in Abb. 12 durch Kabelverbindungen auf zentralen auswechselbaren Programmierfeldern, in Abb. 13 dagegen in einem nicht auswechselbaren Programmierfeld allein durch

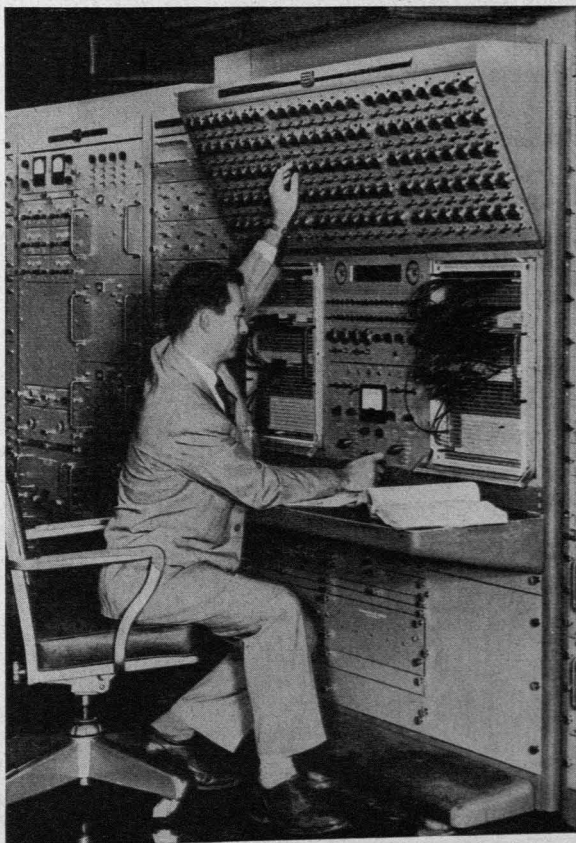


Abb. 12. Ausschnitt aus der Analogie-Rechenanlage PACE der Electronic Associates, Rechenzentrum in Brüssel. Oben: Koeffizienten-Potentiometer, darunter Schalt- und Meßeinrichtungen mit zwei auswechselbaren Programmierfeldern, unten Spannungsversorgung, Recheneinheiten. Foto Electronic Associates, Inc.

Kurzschlußstecker, wobei Steckschablonen für häufig auszuführende Steckerkonfigurationen vorhanden sind. Durch Wahlschalter können die Ausgänge aller operativen Recheneinheiten an Anzeigeeinheiten gegeben werden. In Abb. 11 sind zur Anzeige mit Kathodenstrahlröhren zwei Einstrahlröhren vorgesehen, in der Anlage Abb. 13 wird eine Zweistrahlröhre (hier mit aufgesetzter Registrierkamera) verwendet. Die auf den Oszillogrammen ange-

zeigten Spannungen können ausgemessen werden. Zur Ausstattung der Anlage Abb. 12 gehört ein elektronisches digitales Anzeige-Voltmeter, das (bei mäßiger Rechengeschwindigkeit) fortlaufend Spannungswerte vierziffrig mit Komma anzeigt. Bei allen drei Anlagen können zusätzliche Registriereinrichtungen (Kathodenstrahl-Oszillographen oder servogesteuerte elektromechanische

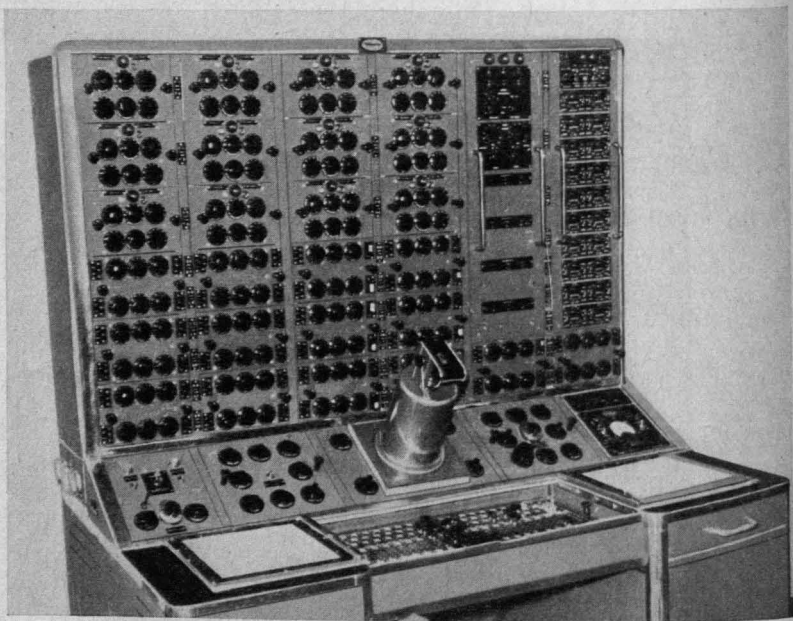


Abb. 13. SHORTS GENERAL PURPOSE ANALOGUE COMPUTER, Gehäuse in Form eines Schreib-  
tisches mit Aufbau. Im Aufbau lineare Funktionseinheiten, Koeffizienteneinheiten, Multiplizierer, Funktions-  
generator. Schräg angeordnet: Schalt- und Meßeinheiten, Anzeige mit Zweistrahlröhre — hier mit auf-  
gesetzter Registrierkamera. Waagrecht: Programmierfeld. Spannungsversorgung im linken, Bandschreiber  
im rechten Seitenschrank.

schreibende Registriergeräte in Trommel- oder Tafelform) angeschlossen werden.

Nach Wahl zeigt die Anlage die Rechenergebnisse repetierend mit Rechenzeiten von 0,1 s bis 10 s (100 s) oder einmalig mit Rechenzeiten von (etwa) 2 s bis 2 Stunden an, so daß die Rechnungen mit Zeitraffung, in natürlicher Zeit und mit Zeitdehnung ausgeführt werden können.

## 2. Programmierung für eine lineare Differentialgleichung

### Ermitteln von Ausgangsprogrammierungen

Beim Rechenvorgang in der Maschine ist die Rechenzeit  $\tau$  die unabhängige Veränderliche; sie ist nach oben beschränkt:  $0 \leq \tau \leq \bar{\tau}$ . Abhängige Veränderliche sind Spannungen, die durch die Überlastungsgrenzen der Recheneinheiten  $\pm U$  beiderseits beschränkt sind,  $-U \leq v, w, \dots \leq U$ . Man kann

mit der Maschine nur solche Aufgaben behandeln, die sich maschinengerecht so umformen lassen, daß die interessierenden Werteverläufe in den Anzeigebereich der Maschine fallen. Die einfachste Umformung besteht darin, daß man Maschinenveränderliche proportional zu den Problemveränderlichen einführt.

Die Differentialgleichung dritter Ordnung mit positiven konstanten Koeffizienten

$$a_3 y''' + a_2 y'' + a_1 y' + a_0 y = f(t) \quad (2.1)$$

für eine Funktion  $y(t)$  der unabhängigen Veränderlichen  $t$  sei für die Anfangsbedingungen  $y(0) = y_0$ ,  $y'(0) = y'_0$ ,  $y''(0) = y''_0$  zu lösen. Die Ableitungen nach  $t$  werden durch Striche bezeichnet. Die Funktion  $f(t)$  wird Anregungsfunktion genannt.

Durch Einführen von *Maschinenveränderlichen*

$$\tau = \kappa t, \quad u = \sigma y, \quad \kappa \dot{u} = \sigma y' \quad (2.2)$$

und Anbringen eines Verfügungsfaktors  $\varrho$  geht (2.1) über in die *Maschinengleichung* (Differentialgleichung in Maschinenveränderlichen)

$$\varrho \kappa^3 a_3 \ddot{u} + \varrho \kappa^2 a_2 \dot{u} + \varrho \kappa a_1 u + \varrho a_0 u = \varrho \sigma f\left(\frac{\tau}{\kappa}\right), \quad (2.3)$$

in der die Ableitungen nach der Maschinenzeit  $\tau$  durch Punkte bezeichnet sind. Die Maschinengleichung ist nun durch ein Rechnetzwerk zu realisieren. Aus der Anzeige der Ergebnisse in Maschinenveränderlichen  $\tau$  und  $u$  sind dann nach (2.2) die entsprechenden Aussagen in den Problemveränderlichen  $t$ ,  $y$  zu erhalten. Für das Herstellen des Rechnetzwerkes wird eine *Programmierskizze* entworfen, die die zu verwendenden Arten und Anzahlen von Recheneinheiten vollständig beschreibt und Angaben über die Einstellwerte für die Widerstände, Zeitkonstanten und die Koeffizienteneinheiten enthält.

Abb. 14 zeigt eine Programmierskizze mit zwei Summierern und einer Integriererkette aus drei Integriern für die Maschinengleichung (2.3). Dabei sollen Integrierer verwendet werden, mit denen die Rechnung beim Anfangswert 0 beginnt. Falls für die Lösung der Differentialgleichung (2.1) Anfangswerte  $y_0 \neq 0$ ,  $y'_0 \neq 0$ ,  $y''_0 \neq 0$  gegeben sind, müssen daher nach Abb. 9c, d Anfangswerte in die nachfolgenden Recheneinheiten eingegeben werden. Die Ausgangsspannungen der Recheneinheiten werden mit  $v$ ,  $v_0$ , ...,  $v_3$  bezeichnet. Die Zeitkonstanten der Integrierer seien  $R_i C_i = T_i$ ,  $i = 1, 2, 3$ .

Der Entwurf der Programmierskizze wird wie folgt motiviert. Um die niederen Ableitungen durch Integrieren zu erhalten, setzt man  $v_0 = \ddot{u}$ . Dann gilt für die Ausgangsspannung  $v_1$  des ersten Integrierers  $\dot{v}_1 = -\frac{v_0}{T_1}$ .

Wegen  $v_1(0) = 0$  wird daher

$$v_1 = -\frac{1}{T_1} \int_0^v v_0 d\tau = -\frac{1}{T_1} \int_0^{\tau} \ddot{u} d\tau = -\frac{\dot{u} - \dot{u}_0}{T_1}.$$



Einführen von  $v$ ,  $v_0$  und  $v_2$  ergibt nun

$$\varrho \kappa^3 a_3 \ddot{u} + \varrho \kappa^2 a_2 (\ddot{u} - \ddot{u}_0) + \varrho \kappa a_1 (\dot{u} - \dot{u}_0) + \varrho a_0 (u - u_0) + w_0 - \varrho \sigma f\left(\frac{\tau}{\kappa}\right) = 0.$$

Hiernach wird die geforderte Maschinengleichung (2.3) erhalten, wenn die Anfangsspannung  $w_0$  durch

$$w_0 = \varrho \kappa^2 a_2 \ddot{u}_0 + \varrho \kappa a_1 \dot{u}_0 + \varrho a_0 u_0 = \varrho \sigma (a_2 y_0'' + a_1 y_0' + a_0 y_0) \quad (2.4)$$

bestimmt wird. (Bei Eingeben der Anfangswerte vor Beginn der Rechnung nach Abb. 9a, b werden  $v_1 = -\frac{\ddot{u}}{T_1}$ ,  $v_2 = \frac{\dot{u}}{T_1 T_2}$ ,  $v_3 = -\frac{u}{T_1 T_2 T_3}$  und  $w_0$  entfällt).

Hiermit ist zunächst eine formal richtige Programmierskizze für die geforderte Maschinengleichung erhalten, in der jeder Wahl der Zeitkonstanten  $T_1$ ,  $T_2$ ,  $T_3$  für die Integrierer bei jeder Wahl für den Funktionsmaßstab  $\sigma$  und den Zeitmaßstab  $\kappa$  wohlbestimmte durch Einstellformeln gegebene Koeffizienteneinstellwerte und einzustellende Anfangswerte entsprechen. In der Gesamtheit der formal der Maschinengleichung entsprechenden Dimensionierungen der Recheneinheiten sind auch die in der Maschine *einstellbaren* Dimensionierungen enthalten. Unter diesen ist diejenige Dimensionierung der Einstellwerte zu ermitteln, mit der (für die gewählte Programmierskizze) das genaueste mit der Maschine zu erhaltende Ergebnis gewonnen werden kann. Es gilt also, aus der Mannigfaltigkeit der verfügbaren Freiheiten für

$$\kappa, \sigma, \varrho, T_1, T_2, T_3$$

diejenigen zu finden, mit denen die Aufgabe möglichst genau gelöst wird. Die genaueste mit der Maschine zu erhaltende Lösung ist bei zweckmäßiger Wahl des Zeitmaßstabes dadurch gekennzeichnet, daß alle Recheneinheiten *voll angesteuert* werden. Dies bedeutet, daß in jeder Recheneinheit während der Rechenzeit mindestens einmal die Grenze der Belastbarkeit eben erreicht, jedoch nicht überschritten wird.

Man kann, wie aus dem folgenden zu entnehmen ist, eine optimale Dimensionierung für eine formale Programmierskizze von vornherein angeben, wenn die Lösung der Differentialgleichung explizit oder tabuliert bekannt ist. Andernfalls ist man genötigt, die *optimale* Programmierung durch Auswerten einer nicht optimalen *Ausgangsprogrammierung* mit Hilfe der Maschine zu ermitteln. Als Ausgangsprogrammierung ist hierzu jede Programmierung geeignet, die eine überlastungsfreie, nicht allzu geringe und nicht allzu ungleichmäßige Aussteuerung der Recheneinheiten ergibt. Im folgenden werden zunächst Verfahren angegeben, mit denen solche *Ausgangsprogrammierungen* erhalten werden können. Im nächsten Abschnitt wird dann die Frage optimaler Programmierungen allgemeiner erörtert.

**1. Verfahren:** Der erfahrene Rechner wählt zunächst eine ihm nach der Aufgabenstellung zweckmäßig erscheinende Rechenzeit, etwa repetierendes Rechnen mit 1 s Rechenzeit, sodann einen Zeitmaßstab, also den Faktor  $\kappa$  in (2.2), sofern dieser nicht durch Rechenzeit und Aufgabenstellung bestimmt



ist. Schließlich wählt er geeignet erscheinende Zeitkonstanten  $T_1, T_2, T_3$ , etwa solche, die sich mit  $C = 0,1 \mu F$  in den Integrierern leicht einstellen lassen. Durch die Einstellformeln für die Koeffizienteneinheiten ergeben sich nun mit passend gewähltem Verfügungsparameter  $\varrho$  einstellbare Werte, die eingestellt werden. Schließlich kann — ohne Änderung an den Zeitkonstanten und den Koeffizienteneinstellungen — der Funktionsmaßstab  $\sigma$  so gefunden werden, daß mindestens eine der Recheneinheiten befriedigend (zu etwa 80 %) ausgesteuert und keine Recheneinheit überlastet ist. Sind alle Recheneinheiten hinreichend gleichmäßig ausgesteuert, etwa zwischen 50 % und 100 %, so ist eine gute Ausgangsprogrammierung erhalten; andernfalls wird die Programmierung — im allgemeinen ohne nochmaliges Ändern von  $\sigma$  — durch *elementare Umformungen* verbessert. Dies sind simultane Änderungen von Zeitkonstanten der Integrierer und Koeffizienteneinstellwerten, so daß die Programmierskizze formal erfüllt bleibt. Wünscht man beispielsweise die Ausgangsspannung  $v_1$  zu verdoppeln, ohne die übrigen Ausgangsspannungen zu ändern, so hat man  $T_1$  zu halbieren,  $T_2$  zu verdoppeln und den von  $v_1$  gespeisten Koeffizienten entsprechend dem neuen Einstellwert umzustellen. Die Aussteuerung für  $v$  wird beeinflusst durch die Größe der beiden gleichen Widerstände  $R$ . Statt dessen kann man die Einstellwerte für die am Eingang des ersten Summierers liegenden Koeffizienten und den am Eingang des zweiten Summierers liegenden Widerstand  $R$  proportional ändern. Sinngemäß ist bei der Lösung einer Differentialgleichung  $n$ -ter Ordnung mit einer Integriererkette der Länge  $n$  vorzugehen.

## 2. Verfahren: Falls Abschätzungen verfügbar sind

$$|y^{(n)}| \leq Y_n; \quad |y^{(j)}| \leq Y_j \text{ bzw. } |y^{(j)} - y_0^{(j)}| \leq Y_j, \quad j = 0, 1, \dots, n-1, \quad (2.5)$$

erhält man eine nach Abb. 15a bzw. b ausgelegte überlastungsfreie Integriererkette mit dem Funktionsmaßstab

$$\sigma = \kappa^n \frac{\eta U}{Y_n}, \quad \eta \leq 1 \quad (2.6)$$

und den rekursiv zu ermittelnden Zeitkonstanten

$$T_1 = \eta_1 \kappa \frac{Y_{n-1}}{Y_n}, \quad \eta_1 \geq \eta; \quad T_2 = \frac{\eta_2}{\eta_1} \kappa \frac{Y_{n-2}}{Y_{n-1}}, \quad \eta_2 \geq \eta; \quad \dots; \quad (2.71)$$

oder

$$T_1 = \eta_1 \kappa \frac{Y_{n-1}}{Y_n}, \quad \eta_1 \geq \eta; \quad T_2 = \frac{\eta_2}{T_1} \kappa^2 \frac{Y_{n-2}}{Y_n}, \quad \eta_2 \geq \eta; \quad \dots; \quad (2.72)$$

oder

$$T_1 = \eta_1 \frac{\sigma Y_{n-1}}{\kappa^{n-1} U}, \quad \eta_1 \geq 1; \quad T_2 = \frac{\eta_2}{T_1} \frac{\sigma Y_{n-2}}{\kappa^{n-2} U}, \quad \eta_2 \geq 1; \quad \dots; \quad (2.73)$$

$$T_j = \frac{\eta_j}{T_1 \dots T_{j-1}} \frac{\sigma Y_{n-j}}{\kappa^{n-j} U}, \quad \eta_j \geq 1.$$

Hierin ist  $U$  die Überlastungsgrenze, und die  $\eta$  sind Verfügungsparameter. Gilt in (2.5) überall für die Extremwerte das Gleichheitszeichen und werden alle Verfügungsparameter  $\eta$  gleich 1 gewählt, so ergibt sich mit (2.6) und (2.71), wie auch mit (2.6) und (2.72), wie auch mit (2.6) und (2.73) eine optimale Integriererkette.

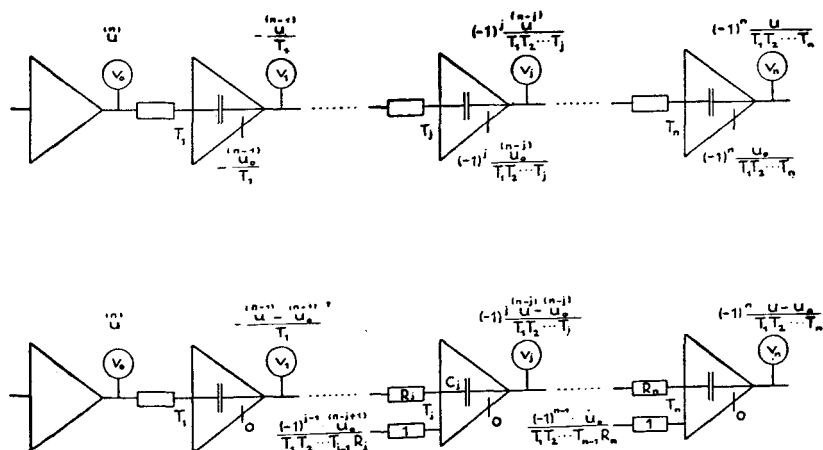


Abb. 15. Reine Integriererketten, a) oben: Eingabe der Anfangswerte im Integrierer selbst, b) unten: Eingabe der Anfangswerte in die nachfolgende Recheneinheit

Zum Beweis hat man zunächst nach (2.2) zwischen den Maschinenveränderlichen und den Problemveränderlichen den Zusammenhang

$$\kappa^j u = \sigma y^{(j)}, \quad j = 0, 1, \dots, n. \quad (2.8)$$

Folglich gelten mit den in Abb. 15 eingeführten Bezeichnungen für die Ausgangsspannungen  $v_j$  die Darstellungen

$$\kappa^{n-j} T_1 T_2 \dots T_j |v_j| = \begin{cases} \sigma |y^{(n-j)}| & \text{(Abb. 15a)} \\ \sigma |y^{(n-j)} - y_0^{(n-j)}| & \text{(Abb. 15b)} \end{cases} \quad (2.9)$$

für  $j = 1, 2, \dots, n$ .

Mit (2.6) folgt daher

$$\kappa^n |v_0| = \sigma |y^{(n)}| = \kappa^n \eta U \frac{|y^{(n)}|}{Y_n},$$

also

$$|v_0| = \eta U \frac{|y^{(n)}|}{Y_n}, \quad |v_0| \leq \eta U, \quad (2.10)$$

und das Gleichheitszeichen wird im Fall  $|y^{(n)}|_M = Y_n$  mit  $\eta = 1$  erreicht.

Mit (2.71), wie auch mit (2.72) ergibt sich zunächst

$$T_1 T_2 \dots T_j = \eta_j \kappa^j \frac{Y_{n-j}}{Y_n}.$$

Daher folgt mit (2.9) und (2.6) — für Abb. 15a —

$$\kappa^{n-j} \eta_j \kappa^j \frac{Y_{n-j}}{Y_n} |v_j| = \kappa^n \frac{\eta}{Y_n} |y^{(n-j)}|,$$

also

$$|v_j| = \frac{\eta}{\eta_j} \frac{U |y^{(n-j)}|}{Y_{(n-j)}}, \quad |v_j| \leq \frac{\eta}{\eta_j} U. \quad (2.10')$$

Schließlich folgt mit (2.73) zunächst

$$\kappa^{n-j} T_1 T_2 \dots T_j = \eta_j \frac{\sigma Y_{n-j}}{U},$$

also mit (2.9)

$$|v_j| = \frac{U}{\eta_j} \frac{|y^{(n-j)}|}{Y_{n-j}}, \quad |v_j| \leq \frac{U}{\eta_j}. \quad (2.10'')$$

Mit Rücksicht auf die angegebenen Beschränkungen für die  $\eta$ ,  $\eta_j$  sind daher die Integriererketten überlastungsfrei. In trivialer Weise ergeben sich die weiteren Aussagen für Integriererketten nach Abb. 15b und für Erreichen optimaler Integriererketten mit optimalen Abschätzungen (2.5).

Nach Wahl von  $\kappa$  und Ermittlung des Funktionsmaßstabes  $\sigma$  und der Zeitkonstanten  $T_j$ ,  $j = 1, 2, \dots, n$  für die Integriererkette erfolgt die Einstellung nach der Programmierskizze. Sind die Abschätzungen (2.5) sehr ungenau, so kann es notwendig werden, die erhaltene Programmierung durch elementare Umformungen in eine befriedigende Ausgangsprogrammierung umzuwandeln.

**3. Verfahren:** Man ermittelt zunächst eine Grundprogrammierung durch die Forderung der *Einstellbarkeit der Koeffizienteneinheiten im Hauptbereich* 0,1 ... 10 sowie der *Zeitkonstanten mit den in den Integrierern verfügbaren Dimensionierungen*. Sodann bestimmt man einen Funktionsmaßstab  $\sigma$ , der für eine Grundprogrammierung geeignet ist. Für die triviale Programmierung zu einer Differentialgleichung  $n$ -ter Ordnung  $\sum_{i=0}^n a_i y^{(i)} = f(t)$  nach dem Schema von Abb. 14 mit Integriererkette nach Abb. 15 werde zunächst  $a_n > 0$ ,  $a_0 \dots a_{n-1} \neq 0$  angenommen. Dann gibt es gewiß einen Verfügungsparameter  $\varrho$ , so daß die Einstellformeln für die Koeffizienteneinstellwerte  $a_{n-j}$

$$\varrho \kappa^n a_n, \dots, \varrho \kappa^{n-j} |a_{n-j}| T_1 \dots T_j, \quad j = 1, 2, \dots, n$$

auf einstellbare Werte führen, wenn die Quotienten aufeinander folgender Einstellwerte hinreichend nahe an 1 sind. Um Einstellbarkeit im Hauptbereich zu erhalten, genügt es, für diese Quotienten zu fordern

$$\sqrt[n]{0,1} \leq \frac{\kappa |a_{n-j+1}|}{|a_{n-j}| T_j} \leq \sqrt[n]{10}, \quad j = 1, 2, \dots, n. \quad (2.11)$$

Dies wird erreicht, wenn die Zeitkonstanten  $T_j$  in den Bereichen gewählt werden:

$$T_j = \varepsilon_j \kappa \frac{|a_{n-j+1}|}{|a_{n-j}|}, \quad \sqrt[n]{0,1} \leq \varepsilon_j \leq \sqrt[n]{10}, \quad j = 1, 2, \dots, n. \quad (2.12)$$

Hierin sind die  $\varepsilon_j$  in dem angegebenen Bereich frei verfügbar, und man erhält einen mehr oder weniger ausgedehnten Bereich von Werten für den Zeitmaßstabsfaktor  $\varkappa$ , so daß sich nach (2.12) einstellbare Zeitkonstanten ergeben. Nachdem  $\varkappa$  und die  $T_j$ ,  $j = 1, 2, \dots, n$  ermittelt sind, werden die Einstellwerte für die Koeffizienten errechnet. Für den Koeffizienten  $a_{n-j}$  erhält man mit (2.12) den Einstellwert

$$\varrho \varkappa^{n-j} |a_{n-j}| T_1 \dots T_j = \varrho \varkappa^n a_n \varepsilon_1 \dots \varepsilon_j,$$

und mit

$$0,1 \leq \left(\sqrt[n]{0,1}\right)^j \leq \varepsilon_1 \dots \varepsilon_j \leq \left(\sqrt[n]{10}\right)^j \leq 10, \quad j = 1, 2, \dots, n$$

folgt, daß beispielsweise mit  $\varrho$  aus  $\varrho \varkappa^n a_n = 1$  alle Koeffizienteneinstellwerte in den Hauptbereich fallen. Zu der hiermit bestimmten Einstellung gibt es gewiß ein so kleines  $\sigma$ , daß an keiner der Recheneinheiten Überlastung eintritt. Man wählt für eine Ausgangsprogrammierung  $\sigma$  so, daß mindestens eine der Recheneinheiten befriedigend angesteuert ist und verbessert die Programmierung, falls notwendig, durch elementare Umformungen.

Sind nicht alle Koeffizienten  $a_i$  von 0 verschieden, so bildet man aus (2.12) formal Produkte, in denen nur vorhandene Koeffizienten stehenbleiben und verwendet mehrfach die gleiche Zeitkonstante. Fehlen beispielsweise die Koeffizienten zwischen  $a_{n-j_1}$  und  $a_{n-j_2}$ , ist also

$$a_{n-j_1} \neq 0, \quad a_{n-j} = 0, \quad a_{n-j_2} \neq 0, \quad j_1 < j < j_2,$$

so erhält man aus (2.12)

$$T_{j_1+1} \dots T_{j_2} = \varepsilon_{j_1+1} \dots \varepsilon_{j_2} \varkappa^{j_2-j_1} \frac{|a_{n-j_1}|}{|a_{n-j_2}|}$$

und setzt die  $j_2 - j_1$  zu ermittelnden Zeitkonstanten gleich groß an gemäß

$$T_j = \varepsilon \varkappa \sqrt[j_2-j_1]{\frac{|a_{n-j_1}|}{|a_{n-j_2}|}}, \quad \sqrt[n]{0,1} \leq \varepsilon \leq \sqrt[n]{10}, \quad j = j_1 + 1, \dots, j_2.$$

Bei linearen Differentialgleichungen mit konstanten Koeffizienten ist die optimale Programmierung für eine nach dem Schema von Abb. 14 (mit Integriererkette nach Abb. 15) entworfene Programmierskizze im wesentlichen durch den Funktionsmaßstab  $\sigma$  und die Zeitkonstanten in der Integriererkette bestimmt, da man über die Aussteuerung des Eingangssummiereis in trivialer Weise verfügen kann. Zur Anwendung des im folgenden Abschnitt geschilderten Verfahrens, mit Hilfe einer überlastungsfreien Ausgangsprogrammierung eine optimale Programmierung zu erhalten, werden an den operativen Recheneinheiten die *Extremwerte*  $|v_j|_M$  der Ausgangsspannungen gemessen.

### 3. Optimale reine Integriererketten

Eine nach einer Programmierskizze erhaltene Einstellung der Zeitkonstanten der Integrierer, der Koeffizienteneinheiten und der Anfangswerte ist – für fest gewähltes  $\varkappa$  und festbleibende Rechenzeit – optimal, wenn es nicht – für fest gewähltes  $\varkappa$  und festbleibende Rechenzeit – möglich ist, durch irgendeine Änderung der Einstellungen (also auch durch Änderung des Funktionsmaßstabes) die Aussteuerung wenigstens einer

Recheneinheit zu verbessern. Eine Recheneinheit kann höchstens „voll“ angesteuert werden, d. h. bis zu Extremwerten  $\pm U$ . Es gibt Programmierskizzen, nach denen es unmöglich ist, alle im Rechnernetzwerk verwendeten Recheneinheiten voll auszusteuern. Im allgemeinen ist es aber dann möglich, eine Programmierskizze optimal zu dimensionieren, wenn die in ihr enthaltenen Integriererketten optimal dimensioniert sind. Dies wird für reine Integriererketten (Abb. 15) durch das folgende Verfahren erreicht. Die reinen Integriererketten sind dadurch gekennzeichnet, daß die Integrierer außer Anfangswerten keine anderen Einspeisungen erhalten.

Gegeben sei eine Programmierskizze, die eine reine Integriererkette der Länge  $n$  enthalte. Diese entspreche einer Problemveränderlichen  $y(t)$  nach dem Ansatz

$$\tau = \kappa t, \quad u = \sigma y. \quad (3.1)$$

Für die gesamte Programmierskizze liege eine überlastungsfreie Ausgangsprogrammierung vor, so daß für die  $y(t)$  entsprechende Integriererkette die Daten  $\sigma, T_1, T_2, \dots, T_n$  gegeben sind. Die Extremwerte der Ausgangsspannungen  $|v_j|_M$  werden gemessen. Gesucht werden für festgehaltenen Zeitmaßstab  $\kappa$  und festbleibende Wahl der Rechenzeit neue Bestimmungsdaten  $\sigma^*, T_1^*, T_2^*, \dots, T_n^*$ , so daß — mit den entsprechenden Koeffizienten- und Anfangswerteinstellungen — Aussteuerungen gemäß

$$|v_j^*|_M = U_j, \quad U_j \leq U, \quad j = 0, 1, \dots, n \quad (3.2)$$

erreicht werden. Hierbei sind  $\pm U$  die Überlastungsgrenzen, und die  $U_j$  werden mitunter zur Vereinfachung der Rechentechnik kleiner als  $U$  gewählt. Die hiermit gekennzeichnete Aufgabe wird durch das folgende Formelsystem gelöst:

*Neuer Funktionsmaßstab:*

$$\sigma^* = \frac{\eta U}{|v_0|_M} \sigma, \quad \eta \leq 1 \quad (3.3)$$

*Neue Zeitkonstanten, rekursiv:*

$$T_1^* = \eta_1 \frac{|v_1|_M}{|v_0|_M} T_1, \quad \eta_1 \geq \eta; \quad T_j^* = \frac{\eta_j}{\eta_{j-1}} \frac{|v_j|_M}{|v_{j-1}|_M} T_j, \quad \eta_j \geq \eta, \quad j = 2, \dots, n \quad (3.41)$$

oder

$$T_1^* = \eta_1 \frac{|v_1|_M}{|v_0|_M} T_1, \quad \eta_1 \geq \eta; \quad T_j^* = \eta_j \frac{|v_j|_M}{|v_0|_M} \frac{T_1 \dots T_{j-1}}{T_1^* \dots T_{j-1}^*} T_j, \quad \eta_j \geq \eta, \quad j = 2, \dots, n \quad (3.42)$$

oder

$$T_1^* = \eta_1 \frac{|v_1|_M}{U} \frac{\sigma^*}{\sigma} T_1, \quad \eta_1 \geq 1; \quad T_j^* = \eta_j \frac{|v_j|_M}{U} \frac{\sigma^*}{\sigma} \frac{T_1 \dots T_{j-1}}{T_1^* \dots T_{j-1}^*} T_j, \quad \eta_j \geq 1, \quad (3.43)$$

$j = 2, \dots, n.$

Zum Beweis hat man die bereits in (2.9) angeschriebenen Eigenschaften der Integriererketten für die Ausgangsprogrammierung und für die neue Dimensionierung heranzuziehen. Dann folgt — für Integriererketten nach Abb. 15a (oben) — mit (3.3)

$$\kappa^n v_0^* = \sigma^* y^{(n)} = \frac{\sigma^*}{\sigma} \sigma y^{(n)} = \frac{\eta U}{|v_0|_M} \kappa^n v_0,$$

also

$$v_0^* = \frac{v_0}{|v_0|_M} \eta U, \quad |v_0^*|_M = \eta U.$$

Mit (2.9) ergibt sich entsprechend

$$\kappa^{n-j} T_1^* \dots T_j^* |v_j^*| = \sigma^* |y^{(n-j)}| = \frac{\sigma^*}{\sigma} \sigma |y^{(n-j)}| = \frac{\sigma^*}{\sigma} \kappa^{n-j} T_1 \dots T_j |v_j|,$$

also

$$v_j^* = \frac{\sigma^*}{\sigma} \frac{T_1 \dots T_j}{T_1^* \dots T_j^*} v_j. \quad (3.5)$$

Nun ergibt sich aus (3.41), wie auch aus (3.42)

$$\frac{T_1 \dots T_j}{T_1^* \dots T_j^*} = \frac{|v_0|_M}{\eta_j |v_j|_M},$$

also folgt mit (3.5) und (3.3)

$$v_j^* = \frac{\eta U}{|v_0|_M} \cdot \frac{|v_0|_M}{\eta_j |v_j|_M} v_j,$$

und hieraus

$$v_j^* = \frac{v_j}{|v_j|_M} \frac{\eta}{\eta_j} U, \quad |v_j^*|_M = \frac{\eta}{\eta_j} U.$$

Schließlich ergibt sich aus (3.43)

$$\frac{T_1 \dots T_j}{T_1^* \dots T_j^*} = \frac{\sigma}{\sigma^*} \frac{U}{\eta_j |v_j|_M},$$

also folgt mit (3.5)

$$v_j^* = \frac{v_j}{|v_j|_M} \frac{U}{\eta_j}, \quad |v_j^*|_M = \frac{U}{\eta_j}.$$

Damit sind alle Behauptungen formal bewiesen. Für Integriererketten nach Abb. 15b (unten) hat man sinngemäß  $y^{(i)}$  durch  $y^{(i)} - y_0^{(i)}$  zu ersetzen. Die in den Herleitungen verwendeten Eigenschaften der Integriererketten sind jedoch beim Rechnen mit realen Recheneinheiten nur genähert erfüllt (vgl. Anhang). Deshalb dürfen die Verbesserungsformeln nicht unmittelbar auf Ausgangsprogrammierungen mit mangelhafter Aussteuerung der Recheneinheiten angewendet werden. Erfahrungsgemäß führen die Verbesserungsformeln zum gewünschten Erfolg, wenn keine der Recheneinheiten in der Ausgangsprogrammierung zu weniger als  $0,1 U$  angesteuert ist. In schwierigen Fällen ist es zweckmäßig, zunächst nur Aussteuerung zu  $U$ ;  $\approx 0,8 U$  anzustreben und dann nochmals zu verbessern. Man begnügt sich mit Aussteuerung  $U$ ;  $\approx 0,8 U$ , wenn nach der Aufgabenstellung gewünscht wird, befriedigende Auswertungen für Bereiche von Koeffizienten oder Anfangswerten zu erhalten.

Die Verbesserungsformeln (3.3), (3.41), (3.42) enthalten die Daten einer einzelnen Integriererkette und sind daher auf jede reine Integriererkette in einem beliebig umfangreichen Rechnernetzwerk anwendbar. Enthält dieses etwa Darstellungen der Funktionen  $y_j$ ,  $j = 1, 2, \dots, k$ , und zugeordnete

Integriererketten der Längen  $n_j$ ,  $j = 1, 2, \dots, k$ , so ist der Formalismus auf alle diese Ketten anzuwenden. Dabei ist allerdings vorausgesetzt, daß die Integriererketten außer Anfangswerten keine Einspeisungen erhalten. Über die Entstehungsweise der  $^{(n)}u$  waren jedoch keine Voraussetzungen notwendig. Insbesondere darf der zugeordnete Summierer Eingänge aus nichtlinearen Recheneinheiten enthalten. Mit den neuen Bestimmungsdaten der Integriererketten werden die Einstellwerte für die Koeffizienteneinheiten und die Anfangswerte sowie die Eingänge für die Anregungsfunktionen nach der Programmierskizze neu eingestellt. Dann sind die Integriererketten wunschgemäß (bis optimal) ausgesteuert, und es zeigt sich keine Überlastung, wenn die übrigen im Netzwerk vorhandenen Recheneinheiten überlastungsfrei eingestellt sind. Summierer, wie beispielsweise in Abb. 14 links, lassen sich stets in trivialer Weise wunschgemäß aussteuern; nichtlineare Recheneinheiten werden grundsätzlich so in die Programmierskizze eingegliedert, daß sie Überlastung am Ausgang nur bei Überlastung am Eingang ergeben können.

Bei der Auswertung von Programmierskizzen für nichtlineare Differentialgleichungen kann der Fall eintreten, daß es bei optimaler Aussteuerung aller linearen Recheneinheiten nicht möglich ist, zugleich alle nichtlinearen Recheneinheiten voll auszusteuern. Tritt dies in einer Programmierung für nur eine nichtlineare Recheneinheit ein, so ist die zugehörige optimale Programmierung eindeutig — mit voller Aussteuerung für die lineare und größtmöglicher Aussteuerung für jene nichtlineare Recheneinheit. Es ist jedoch denkbar, daß mehrere nichtlineare Recheneinheiten nicht voll ausgesteuert werden können und daß dann Einstellungen mit verschiedenen Begrenzungen für die Aussteuermöglichkeit vorliegen. In solchen Fällen hat man zu beurteilen, welche der Möglichkeiten die genaueste Auswertung liefert. Die Frage der Beurteilung wird später erörtert.

#### 4. Mannigfaltigkeiten von Programmierskizzen

In den vorangehenden Erörterungen wurde von einer für eine Aufgabe entworfenen Programmierskizze ausgegangen und für diese eine optimale Programmierung ermittelt. Bis auf die einfachsten Aufgaben ist jedoch die Programmierskizze keineswegs eindeutig durch die Aufgabenstellung festgelegt. Vielmehr erhält man eine mehr oder weniger umfangreiche Mannigfaltigkeit von Programmierskizzen für die gleiche Aufgabe, indem man entweder vor der Programmierung algebraische Umformungen, analytische Umformungen oder Transformationen der unabhängigen oder (und) der abhängigen Veränderlichen vornimmt oder auch aus bereits ausgearbeiteten Programmierskizzen durch Strukturumformungen neue Programmierskizzen herleitet. Dies werde an zwei einfachen Beispielen gezeigt.

Nach dem vorangehenden kann man Abb. 14 als eine zur Aufgabe (2.1) gehörige „triviale“ Programmierskizze bezeichnen. Dort werden die von  $v_1$  und  $v_3$  stammenden Einspeisungen über zwei Umkehrer an den ersten Integrierer geleitet, man kann sie daher vorzeichenrichtig auch unmittelbar in den ersten Integrierer eingeben und erhält durch entsprechendes Umprogrammieren die neue Programmierskizze Abb. 16. Hier ist die Anregungsfunktion durch bloßes Ändern des Vorzeichens trivial verlagert. Die neuen

Koeffizienteneinstellwerte sind aus den in Abb. 14 verzeichneten erhalten, indem diese, der Überführung entsprechend, durch  $\varrho \kappa^3 a_3 R_1$  dividiert sind. Bei dieser Strukturumformung fällt der erste Summierer weg und  $v_0$  erhält eine andere Bedeutung; denn nun wird wegen der Eigenschaft des Summierers (Abb. 7)

$$\varrho \kappa^3 a_3 v_0 + \varrho \kappa a_1 (\dot{u} - \dot{u}_0) + w_0 - \varrho \sigma f\left(\frac{\tau}{\kappa}\right) = 0. \quad (4.1)$$

Nach Abb. 8 gilt für den ersten Integrierer mit den in Abb. 16 eingetragenen Einstellwerten für die Koeffizienten

$$\frac{\ddot{u}}{T_1} = \frac{v_0}{T_1} - \frac{a_2 (\ddot{u} - \ddot{u}_0)}{\kappa a_3 R_1 C_1} - \frac{a_0 (u - u_0)}{\kappa^3 a_3 R_1 C_1}, \quad (4.2)$$

so daß sich durch Einführen von (4.1) und (2.4) die zu realisierende Maschinen-gleichung (2.3) ergibt. In der Programmierskizze nach Abb. 16 wird  $y'''$  nicht

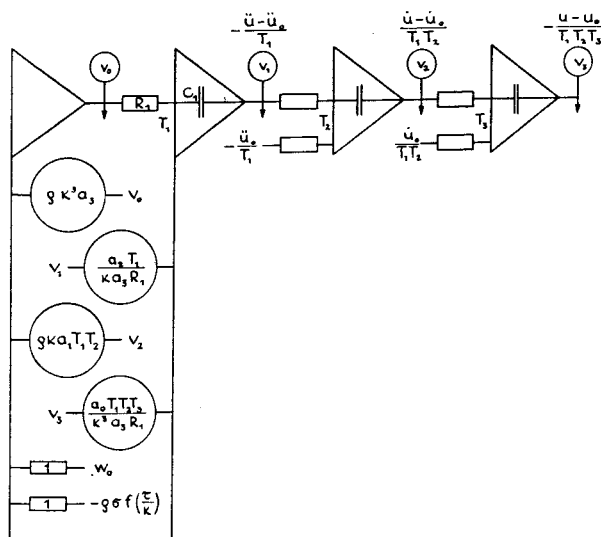


Abb. 16. Weitere Programmierskizze, aus Abb. 14 durch Umprogrammieren hergeleitet

angezeigt. Wünscht man Anzeige, so kann  $\ddot{u}$  in einer Nebenrechnung nach (4.2) mit einem Summierer erzeugt werden. Die Integriererkette in Abb. 16 ist keine „reine“ Integriererkette, da hier der erste Integrierer andere Einspeisungen als Anfangswerte erhält. Die Verbesserungsformeln aus Abschnitt 3 sind daher hier nicht ohne weiteres anwendbar. Ist jedoch eine optimale Programmierung für Abb. 14 verfügbar, so sind auch nach der Umprogrammierung die Ausgänge für die Integrierer optimal, da  $v_1$ ,  $v_2$ ,  $v_3$  nicht geändert werden; lediglich am Summierer kann eine triviale Änderung der Einstellungen notwendig werden, um Übersteuerung zu vermeiden oder optimale Aussteuerung zu erreichen.



Im folgenden wird an einem Beispiel einer nichtlinearen Differentialgleichung gezeigt, daß die Mannigfaltigkeit der Programmierskizzen zur Lösung einer Aufgabe recht umfangreich sein kann, und daß die Genauigkeit der mit verschiedenen optimalen Programmierungen erhaltenen Lösungen sehr verschieden ausfallen kann, auch wenn es sich um Programmierskizzen mit gleicher minimaler Anzahl von Recheneinheiten handelt.

*Die van der POLsche Differentialgleichung*

$$y'' + \mu (y^2 - 1) y' + y = f(t), \quad y(0) = y_0 = 0, \quad y'(0) = y'_0 = 1 \quad (4.3)$$

geht durch Einführen von Maschinenveränderlichen nach (2.2) und Multiplizieren mit einem Verfügungsparameter  $\varrho$  über in die Maschinengleichung

$$\varrho \kappa^2 \ddot{u} + \mu \left( \frac{u^2}{\sigma^2} - 1 \right) \varrho \kappa \dot{u} + \varrho u = \varrho \sigma f \left( \frac{\tau}{\kappa} \right), \quad u_0 = 0, \quad \kappa \dot{u}_0 = \sigma. \quad (4.4)$$

Mit einem Funktionserzeuger  $F(w) = \frac{2w^2 - U}{U}$  und einem Multiplizierer erhält man hieraus die triviale Programmierskizze Abb. 17a (oben). In die Integriererkette ist ein Umkehrer eingefügt, um den geforderten Anfangswert einzugeben; um alle Freiheiten für die Einstellmöglichkeiten auszunutzen zu können, sind außer  $\varrho$  noch weitere Verfügungsparameter durch Koeffizienten an den Eingängen der nichtlinearen Recheneinheiten eingeführt. Dann erfordert die Identifizierung mit der Maschinengleichung Einstellen der Beziehung

$$\frac{1}{\sigma^2} = \frac{2\lambda_1^2}{T_1^2 T_2^2 U^2}.$$

Durch eine solche Anforderung wird die Möglichkeit, günstige Aussteuerung zu erreichen, erheblich eingeschränkt. Man kann dies vermeiden, indem man die Klammer in (4.3) und sinngemäß in (4.4) ausmultipliziert und erst dann in trivialer Weise programmiert. So ergibt sich die Programmierskizze Abb. 17b (unten), in der bei Änderung von  $\mu$  zwei Koeffizienteneinstellungen zu ändern sind, im übrigen jedoch übersichtliche Voraussetzungen für optimales Programmieren vorliegen. Die Integriererkette ist trotz der Unterbrechung durch einen Umkehrer eine reine Integriererkette, die mit geringfügiger Abwandlung der Verbesserungsformeln aus Abschnitt 3 optimal dimensioniert werden kann. Ist  $v_2$  voll ausgesteuert, so wird zwangsläufig der Quadrierer voll ausgesteuert. Die Möglichkeit, den ersten Summierer voll auszusteuern, ist von den übrigen Recheneinheiten unabhängig. Da sich jedoch die Extremwerte von  $\dot{u}$  und  $u^2$  nicht gleichzeitig einstellen, ist es unmöglich, den Multiplizierer voll auszusteuern. Für kleine  $\mu$  ist der hierdurch hervorgerufene Fehler geringfügig, für große  $\mu$  ist die Programmierskizze völlig unbrauchbar.

Die gekennzeichnete Fehlerquelle kann durch analytische Umformung der Aufgabe vermieden werden. Durch Integrieren und einfache Umformungen ergeben sich aus (4.3) und  $\left(\frac{y^3}{3} - y\right)' = (y^2 - 1)y'$  die folgenden vier Fassungen unserer Aufgabe:

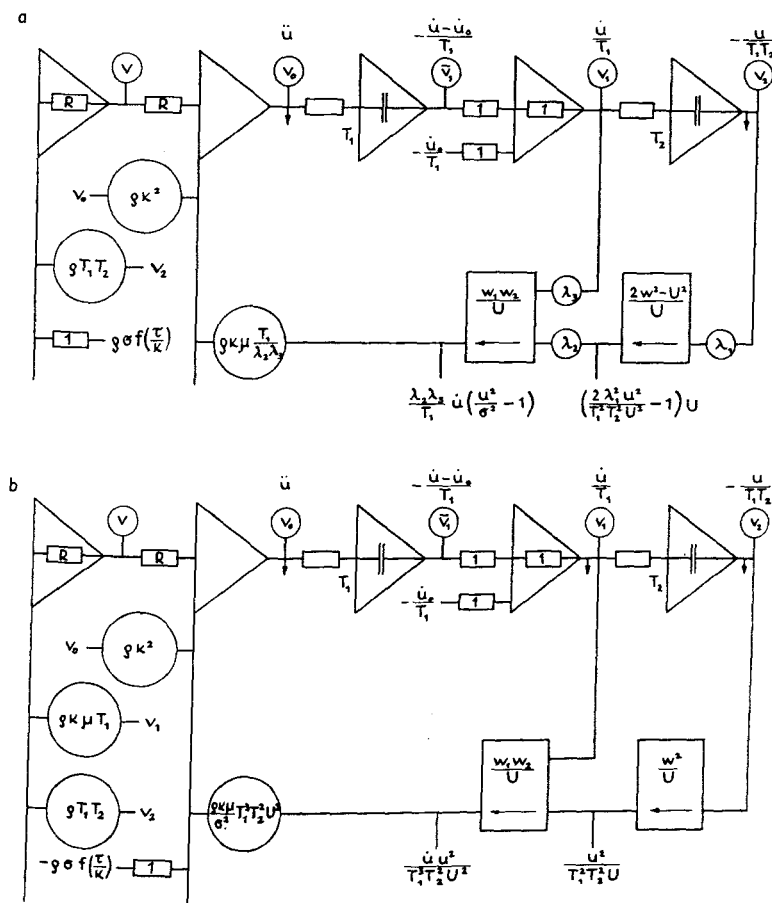


Abb. 17. Differentialgleichung von *van der POL*, a) oben: triviale Programmierung, b) unten: Programmierung nach Ausmultiplizieren der Klammer

$$y' - y'_0 + \mu \left( \frac{y^3}{3} - y \right) + \int y \, dt = 0 \quad (= \int f(t) \, dt) \quad (4.5)$$

$$y' - y'_0 + \mu y \left( \frac{y^2}{3} - 1 \right) + \int y \, dt = 0 \quad (4.6)$$

$$y' - y'_0 + \mu \frac{y^3}{3} - \mu y + \int y \, dt = 0 \quad (4.7)$$

$$y' - y'_0 + \frac{\mu}{3} y \cdot y^2 - \mu y + \int y \, dt = 0. \quad (4.8)$$

Durch jede dieser Schreibweisen ist eine Programmierskizze (im wesentlichen eindeutig) beschrieben. Dabei ist in (4.5) und (4.6) je einmaliges, in (4.7) und

(4.8) je zweimaliges Einstellen von  $\mu$  notwendig. Jede dieser Programmierskizzen kann optimal mit voller Aussteuerung aller zu verwendenden Recheneinheiten realisiert werden. Für (4.8) ergibt sich die Maschinengleichung

$$\varrho \kappa \dot{u} - \varrho \kappa \dot{u}_0 + \frac{\varrho \mu}{3\sigma^2} u \cdot u^2 - \varrho \mu u + \frac{\varrho}{\kappa} \int u d\tau = 0. \quad (4.9)$$

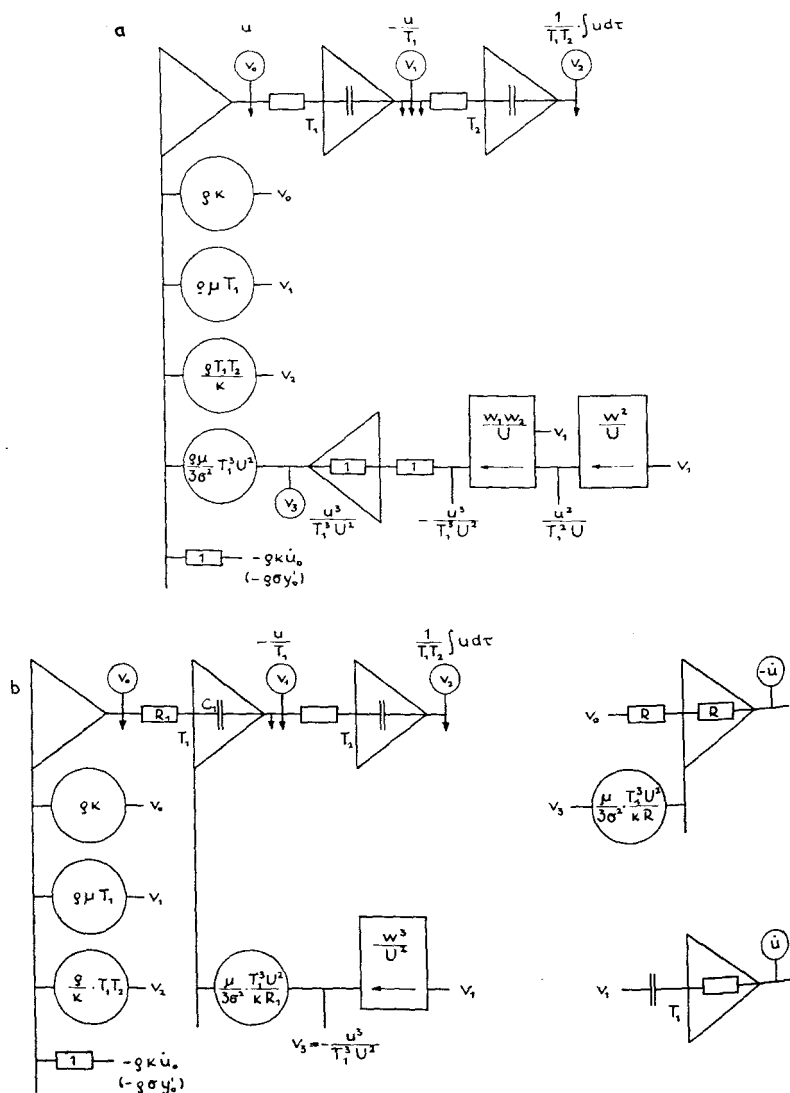


Abb. 18. Programmierungen zu *van der POLs* DGL in integrierter Form, a) oben: triviale Programmierung über  $u_0$  ist  $u$  durch  $\dot{u}$  zu ersetzen, b) unten: nach Strukturumformung

Hiermit erhält man die Programmierskizze Abb. 18a (oben), in der der Quadrierer durch einen Multiplizierer ersetzt werden kann. Diese Programmierung liefert Anzeigen für  $y'$ ,  $y$ ,  $\int y dt$ ,  $y^2$ ,  $y^3$  (Abb. 21) und alle hieraus zu erzielenden Paarungen (Phasendiagramme), von denen etliche in Abb. 22 dargestellt sind. Da hier der Multiplizierer voll angesteuert werden kann, ergeben sich zutreffende Lösungen für einen sehr ausgedehnten Bereich von  $\mu$ -Werten. Mit der schon beim Übergang von Abb. 14 zu Abb. 16 angewendeten Strukturumformung ergibt sich aus Bild 18a (oben) die Programmierskizze Abb. 18b (unten). Bei Verwenden eines Funktionserzeugers für die dritte Potenz ist dies eine Programmierskizze mit geringstem Aufwand an Recheneinheiten, in der die — im Rechnetzwerk selbst vermiedene — Anzeige von  $y'$  in einer Nebenrechnung erreicht werden kann. Mit diesen Beispielen für Programmierskizzen zur van der POLschen Differentialgleichung ist die Mannigfaltigkeit aller für diese Aufgabe möglichen Programmierungen nicht erschöpft; eine umfassendere Auswahl wird im folgenden Abschnitt mit vereinfachter Symbolik beschrieben.

### 5. Mannigfaltigkeiten von Strukturskizzen

Zu jeder Aufgabe gibt es im allgemeinen etliche Programmierskizzen, die sich im wesentlichen eindeutig durch die Schreibweise der Aufgabe und Vereinbarungen über die in der Programmierskizze zu verwendende *Symbolik* kennzeichnen lassen. Dies gilt beispielsweise für Abb. 14 als Programmierskizze zu (2.1) und für Abb. 18a als Programmierskizze zu (4.8); ebenso gibt es zu (4.5), (4.6), (4.7) je eine Programmierskizze, die durch die Schreibweise der Aufgabe mitsamt Angabe der Art der zu verwendenden nichtlinearen Recheneinheiten gekennzeichnet ist. Programmierskizzen, die durch Strukturumformungen oder mit besonderen Kunstgriffen zu erhalten sind, müssen jedoch durch zusätzliche Hinweise charakterisiert werden. Zur Kennzeichnung einer beabsichtigten Programmierskizze genügt eine mit vereinfachten Symbolen angelegte *Strukturskizze*, beispielsweise als Anweisung für Mitarbeiter, die mit der Skizzierweise und der Symbolik vertraut sind.

Einige für Strukturskizzen *vereinfachte Symbole* sind in Abb. 19 zusammengestellt. Auf Darstellung koppelnder Widerstände wird verzichtet. Summierer

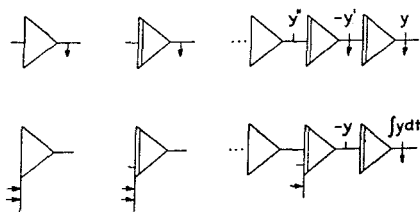


Abb. 19. Beispiele zur vereinfachten Symbolik für Strukturskizzen

werden durch Dreiecke, Integrierer durch Dreiecke mit verdoppelter Basis, nichtlineare Recheneinheiten durch Rechtecke mit Buchstabenzeichen wie in Abb. 10 gekennzeichnet. Für Ausgänge aus Recheneinheiten und für Eingänge

über Koeffizienteneinheiten werden Pfeile gesetzt; dabei müssen in einer vollständigen Skizze Ausgangs- und Eingangspfeile in gleicher Anzahl auftreten. Verbindet ein Pfeil einen Ausgang unmittelbar mit einem Eingang, so wird er sowohl als Ausgangs- wie als Eingangspfeil gezählt. Eingeben von Anfangswerten wird in der Strukturskizze durch einen kurzen Strich angedeutet. Diese Kennzeichnung kann auch entfallen, wenn das Eingeben der Anfangswerte routinemäßig klar ist. Die Strukturskizze wird mit Problemveränderlichen beschriftet, wobei es im allgemeinen genügt, nur einige wenige Eintragungen vorzunehmen, etwa die Ausgänge zu kennzeichnen, für die Registrierung verlangt wird.

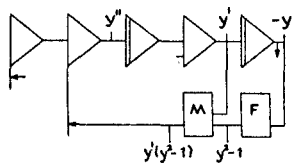
Mit diesen Vereinbarungen erhält man für die Differentialgleichung von van der POL die in Abb. 20 zusammengestellten Strukturskizzen. Der Tabellenkopf enthält vier Schreibweisen der Aufgabe. Darunter stehen in der ersten Zeile die zugehörigen trivialen Strukturskizzen, durch deren erste die Programmierskizze Abb. 17a gekennzeichnet ist. Durch Strukturumformung und Verzicht auf Anzeige der höchsten Ableitung entstehen die Skizzen in der zweiten Zeile. Die dritte Zeile geht aus der ersten hervor. In den beiden links stehenden Skizzen ist der Umkehrer für Eingeben des Anfangswertes  $y_0$  weggefallen, daher muß der Anfangswert an zwei bzw. drei Stellen eingegeben werden. In den beiden rechts stehenden Skizzen ist der Funktionsgenerator in zwei Recheneinheiten zerlegt. Die letzte Zeile geht aus der dritten durch die schon beim Übergang von der ersten zur zweiten angewendete Strukturumformung hervor, links wird auf Anzeige von  $y''$ , rechts auf Anzeige von  $y'$  verzichtet. Durch den Tabellenkopf ist beiläufig darauf hingewiesen, wie diese Ableitungen durch Nebenrechnung mit einem Summierer erhalten werden können. Die Zusammenstellung Abb. 20 ist keineswegs erschöpfend, beispielsweise kann man die Funktionsgeneratoren mit umgekehrtem Vorzeichen einstellen und so einige weitere bemerkenswerte Strukturskizzen erhalten. Ferner sind an einigen Stellen Verschiebungen der Koeffizienteneingabe und der Anfangswerteingabe möglich. Schließlich sind nur unmittelbar naheliegende Umformungen der gestellten Aufgabe berücksichtigt; im allgemeinen sind auch tieferliegende Transformationen der Veränderlichen zu erörtern.

Diskussion und Auswertung aller gekennzeichneten Strukturskizzen gibt die Möglichkeit, eine Fülle verschiedenartiger Vorkommnisse bei guter und schlechter Programmierung zu demonstrieren. Unsere Aufgabe eignet sich daher besonders gut für die Ausbildung. Die trivialen Strukturskizzen erfordern den größten Aufwand an Recheneinheiten. In der ersten und in der dritten Zeile der Zusammenstellung Abb. 20 liegen reine Integriererketten vor, so daß nach dem Verfahren in Abschnitt 3 optimiert werden kann. Optimale Programmierungen für die zweite und die vierte Zeile können dann durch Umprogrammieren erhalten werden. Alle unmittelbaren Strukturskizzen (linke Hälfte) haben den Mangel, daß es mit den zugeordneten Programmierungen nicht möglich ist, den Multiplizierer voll auszusteuern (vgl. Oszillogramme Abb. 21). Hierdurch wächst der Fehler der Ergebnisse mit  $\mu$  an bis zur völligen Unbrauchbarkeit. Nach den zur integrierten Differentialgleichung gehörigen Strukturskizzen (rechte Hälfte) ergeben sich lauter Programmierskizzen, nach denen volle Aussteuerung aller Recheneinheiten möglich ist. Die größte Ergebnisgenauigkeit wird dann mit geringstem Aufwand an Recheneinheiten er-

## GRUNDFORM

$$y'' + \mu y'(y^2 - 1) + y = 0$$

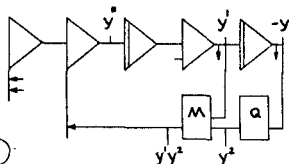
$$y_0 = 0; y'_0 = 1$$



①

## AUSMULTIPLIZIERT

$$y'' + \mu y' y^2 - \mu y' + y = 0$$

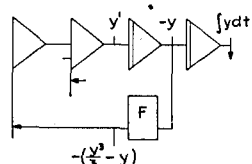


①

## INTEGRIERT

$$y' - y'_0 + \mu \left( \frac{y^3}{3} - y \right) + \int y dt = 0$$

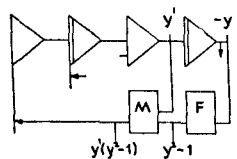
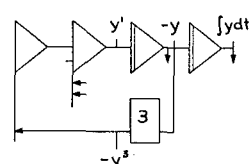
$$y_0 = 0; y'_0 = 1$$



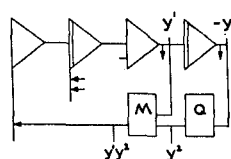
①

## AUSMULTIPLIZIERT

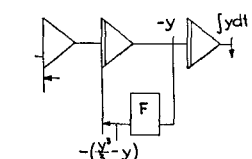
$$y' - y'_0 + \frac{\mu}{3} y^3 - \mu y + \int y dt = 0$$



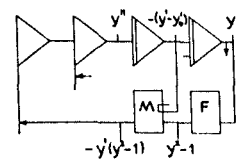
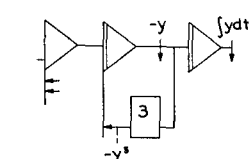
②



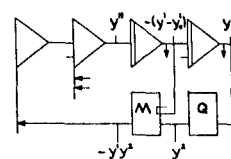
②



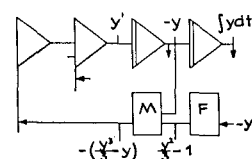
②



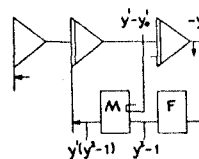
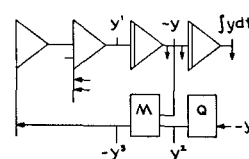
③



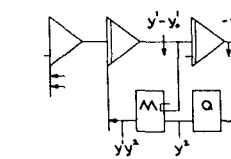
③



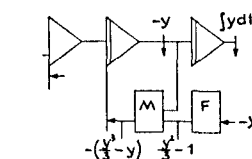
③



④



④



④

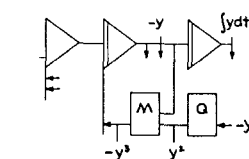


Abb. 20. Strukturskizzen zu van der POLs Differentialgleichung

reicht. Bei der Untersuchung für den Bereich  $0,1 \leq \mu \leq 1200$  zeigt sich besonders für große  $\mu$ -Werte, daß die Ergebnisgenauigkeit auch bei minimalem Aufwand an Recheneinheiten merklich durch die Wahl der Strukturskizze beeinflußt wird.

## 6. Arbeitstechnik beim Rechnen mit Analogie-Rechenmaschinen

Bei Vorliegen einer Aufgabe beginnt man mit der Herstellung einer möglichst umfassenden Übersicht über Strukturskizzen, die der Aufgabe unmittelbar oder nach sachgemäßer Umformung entsprechen. Hiernach beurteilt man,

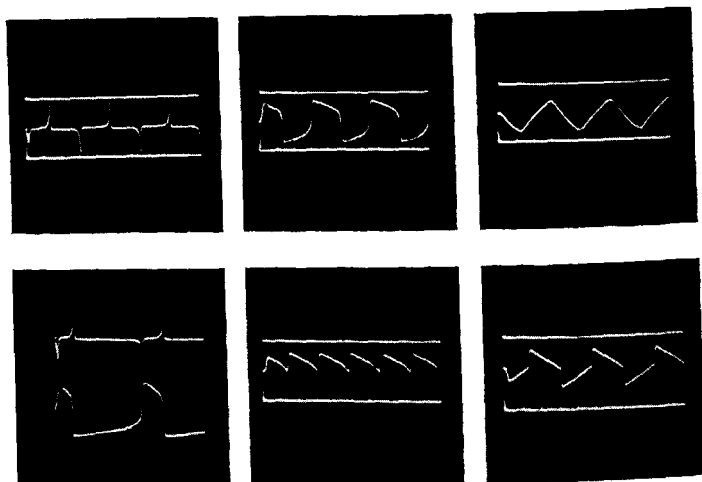


Abb. 21. Oszillogramme zu *van der POLs* Differentialgleichung in der Anordnung  $\begin{pmatrix} y' \\ y \end{pmatrix} \begin{pmatrix} y \\ y^2 \end{pmatrix} y^3 \frac{dy}{dt}$ .

Links unten  $\begin{pmatrix} y' \\ y \end{pmatrix}$  mit formal richtiger, jedoch für große  $\mu$  ungeeigneter Programmierskizze nach Abb. 17b.

in welchem Umfange optimale Programmierungen durch vorhandene Vorschriften oder durch Umprogrammierungen oder durch spezielle Untersuchungen, beispielsweise durch Abschätzungen erhalten werden können. Sodann wählt man eine oder mehrere Strukturskizzen aus, für die Programmierskizzen anzufertigen und auszuwerten sind. Der Rechengvorgang wird dann durch die Wahl des Zeitmaßstabes und der Rechenzeit festgelegt. Hierbei sind die Frequenzeigenschaften der Recheneinheiten und der Registriergeräte zu berücksichtigen, insbesondere beim Aufzeichnen mit schreibenden Registriergeräten. Falls der Verlauf der zu registrierenden Funktionen überhaupt noch nicht bekannt ist, legt man Zeitmaßstab und Rechenzeit erst nach einer orientierenden Ausgangsprogrammierung fest. Hierbei wird beachtet, ob durch die Aufgabenstellung der interessierende Bereich für die unabhängige Veränderliche vorgegeben oder durch Eigenschaften der zu registrierenden Funktionen beschrieben ist. Mitunter ist es zweckmäßig, Auswertungen mit ver-

schiedener Wahl des Zeitmaßstabes zu registrieren. *Außer den Funktionen werden Phasendiagramme registriert* (vgl. Abb. 22). Alle Registrierungen erfolgen mit quantitativen Angaben über die Darstellungsmaßstäbe. Bei Anzeige auf der Kathodenstrahlröhre werden Einzeldaten ausgemessen. Die genauesten Ergebnisanzeigen erhält man mit digitalen Voltmetern und angeschlossenen Tabellendruckern. Schließlich ist die Genauigkeit der erhaltenen Lösung zu beurteilen. Dies geschieht mit Testprogrammen, durch Kontrollrechnungen mit Ziffernverfahren und durch Erwägungen, zu denen man durch den Anblick der Oszillogramme, besonders der Phasendiagramme, angeregt wird.

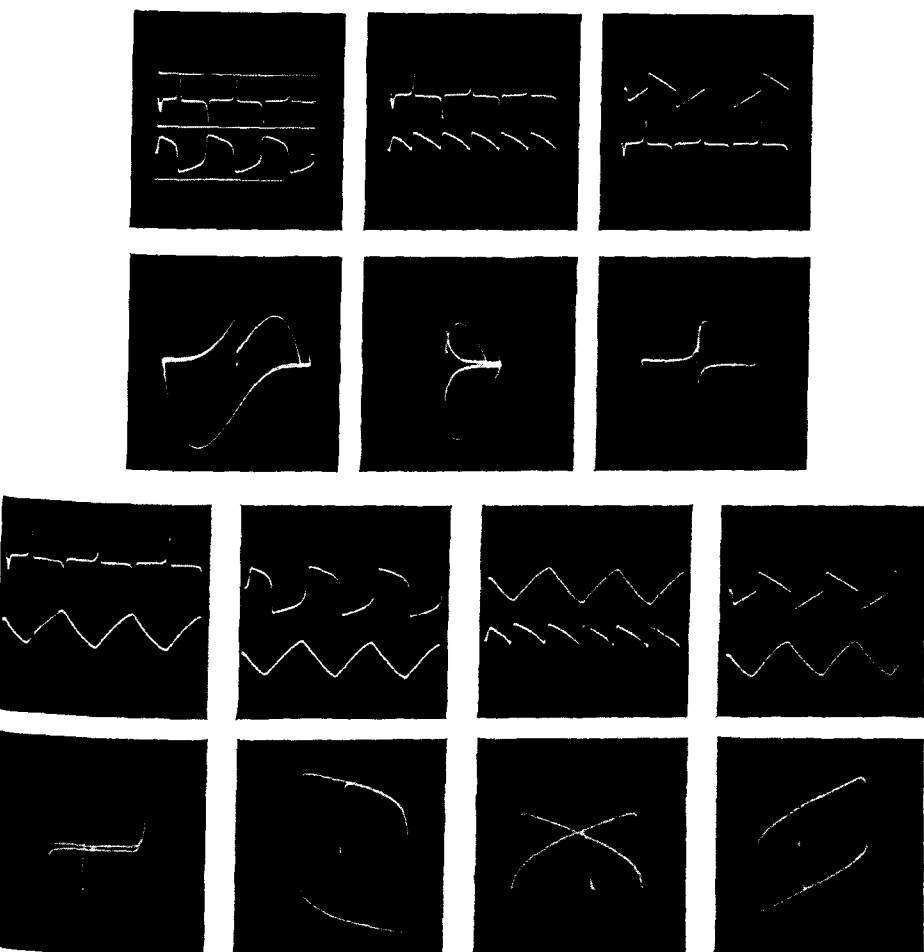


Abb. 22. Paarungen und zugehörige Phasendiagramme: Darstellungen der erzeugten Funktionen in Abhängigkeit voneinander.



## Anhang

## Reale Übertragungseigenschaften linearer operativer Recheneinheiten

Mit den in Abb. 23 angemerkten Bezeichnungen

- $v_1, \dots, v_n$  Eingangsspannungen der Recheneinheit  
 $Y_1, \dots, Y_n, Y_0$  Admittanzen der passiven Zweipole  
 $Y$  (gerichtete) Admittanz des Verstärkers  
 $Y_E$  Verstärkereingangs-Admittanz  
 $Y_A$  Admittanz der Belastung der Recheneinheit  
 $w$  Spannung am Eingang des Verstärkers  
 $u$  Spannung am Ausgang des Verstärkers

ergibt sich aus Abb. 23 für die Spannungen  $w$  am Eingang und  $u$  am Ausgang des Verstärkers nach der Vierpoltheorie bei Vernachlässigung des Gitterstromes der Eingangsrohre der Ansatz

$$\begin{pmatrix} \sum Y_i + Y_E + Y_0 & -Y_0 \\ -Y_0 + Y & Y_0 + Y_A \end{pmatrix} \begin{pmatrix} w \\ u \end{pmatrix} = \begin{pmatrix} \sum Y_i v_i \\ 0 \end{pmatrix}. \quad (1)$$

Hieraus folgt für die Ausgangsspannung

$$u = \frac{-(-Y_0 + Y) \sum Y_i v_i}{Y_0(-Y_0 + Y) + (Y_0 + Y_A)(\sum Y_i + Y_E + Y_0)} \quad (2)$$

Für die Beurteilung von Recheneinheiten und für Genauigkeitsfragen ist die Kenntnis der Verstärkung wesentlich. Für den offenen Verstärker (ohne Rückkopplung,  $Y_0 = 0$ ) und mit nur einem Eingang  $v_1$  liefert (2)

$$u = -\frac{Y_1 Y v_1}{(Y_1 + Y_E) Y_A}.$$

Mit  $Y_1 \gg Y_E$  erhält man daher, wenn die Belastungsimpedanz mit  $Z_A$  bezeichnet wird,

$$u = -\frac{Y}{Y_A} v_1 = -Z_A Y v_1.$$

Daher ist  $A(p) = -Z_A Y$  die Verstärkung. Sie ist frequenzabhängig und kann in dem Frequenzbereich, in dem Anwendungen der Recheneinheiten sachgemäß sind, durch Ansätze der Form

$$A(p) = \frac{A_0}{1 + T_v p} \quad \text{oder} \quad A(p) = A_0 \frac{F_1(p)}{F_2(p)} \quad \text{mit} \quad F_1(0) = F_2(0) = 1 \quad (3)$$

dargestellt werden. Hierin ist  $A_0$  die Verstärkung für die Frequenz 0 und  $T_v$  die größte für den Verstärker charakteristische Zeitkonstante. Für Recheneinheiten werden Verstärker mit großer Verstärkung  $A_0$  verwendet.  $A_0$  liegt zwischen  $10^4$  (für repetierendes Rechnen) und  $3 \cdot 10^8$  (für langdauerndes Rechnen).

Aus (2) folgt durch einfache Umformung

$$u \left( 1 - \frac{Y_0 + Y_A}{Y_0 - Y} \left( 1 + \frac{Y_E}{Y_0} + \sum_{i=1}^n \frac{Y_i}{Y_0} \right) \right) = - \sum_{i=1}^n \frac{Y_i}{Y_0} v_i.$$

Durch Einführen von Impedanzen ( $Y_i Z_i = 1$ ) und mit  $-Y = A(p)$   $Y_A$  ergibt sich hieraus

$$(1 - \varepsilon(p)) u = - \sum_{i=1}^n \frac{Z_0}{Z_i} v_i \quad (4)$$

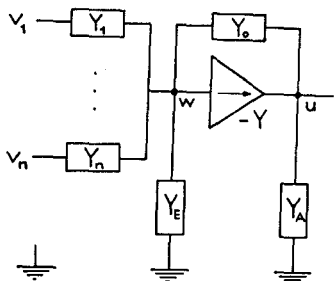
oder

$$u = - \left( 1 + \frac{\varepsilon}{1 - \varepsilon} \right) \sum_{i=1}^n \frac{Z_0}{Z_i} v_i \quad (4')$$

mit

$$\varepsilon(p) = \frac{1 + Y_A Z_0}{1 + Y_A Z_0 A} \left( 1 + \frac{Z_0}{Z_A} + \sum_{i=1}^n \frac{Z_0}{Z_i} \right). \quad (4'')$$

Abb. 23. Lineare operative Recheneinheit



Für hinreichend große Verstärkung wird der Einfluß von  $\varepsilon(p)$  in (4) und (4') nach (4'') vernachlässigbar klein, und für ideale Verstärker mit  $\varepsilon(p) = 0$  folgen aus (4) oder (4') die Programmieransätze

$$u = - \sum_{i=1}^n \frac{Z_0}{Z_i} v_i, \quad \frac{u}{Z_0} + \sum_{i=1}^n \frac{v_i}{Z_i} = 0. \quad (5)$$

Für  $n = 1$  erhält man hieraus (1.1) und (1.2).

Beim Programmieren werden stets die Programmieransätze verwendet. Da jedoch reale Verstärker benutzt werden, sind Genauigkeitsfragen zu erörtern. Mit den Darstellungen (4), (4') für die realen Übertragungseigenschaften der Recheneinheiten kann man die Anwendungsbereiche der Recheneinheiten kennzeichnen, in denen mit den Programmieransätzen eine geforderte Genauigkeit erreicht wird [4], [5]. Praktisch orientiert man sich über die Abweichungen der realen von den idealen Übertragungseigenschaften der Recheneinheiten einer gegebenen elektronischen Analogie-Rechenmaschine durch Tests.

### Literatur

- [1] Korn, G. A. and Korn, Th. M., Electronic Analog Computers, McGraw-Hill, New York, Toronto, London, 1952, 1956.
- [2] Wass, C. A. A., An Introduction to Electronic Analogue Computers. Pergamon Press, London, 1955.
- [3] Johnson, C. L., Analog Computer Techniques, McGraw-Hill, New York, Toronto, London, 1956.
- [4] Macnee, A. B., Some Limitations on the Accuracy of Electronic Differential Analyzers, Proc. I. R. E., Vol. 40 (1952), 303–308.
- [5] Paul, R. J. A., Some Factors Affecting the Accuracy of Electronic Analogue Computers. International Analogy Computation Meeting, Proceedings (1955), 232–238.
- [6] Raymond, F. H., Quelques Remarques sur la Précision des Calculateurs Analogiques, Intern. Anal. Comp. Meeting, 1955, Proceedings, 239–244.
- [7] Peretz, R., Calculateur Analogique Electronique. Le Calcul Analogique au Service de l'Industrie, Brüssel, 1955, 41–45.
- [8] Herrmann, H., Erfahrungen aus dem Probetrieb einer Analogiemaschine. Association Internationale pour le Calcul Analogique, Bulletin d'Information Nr. 3 (1957), 11–13.